NASA Technical Paper 1391

1271

TACT1, A Computer Program for the Transient Thermal Analysis of a Cooled Turbine Blade or Vane Equipped With a Coolant Insert

II - Programmers Manual

Raymond E. Gaugler

JANUARY 1979

NASA

NASA Technical Paper 1391

TACT1, A Computer Program for the Transient Thermal Analysis of a Cooled Turbine Blade or Vane Equipped With a Coolant Insert

II - Programmers Manual

Raymond E. Gaugler
Lewis Research Center
Cleveland, Ohio

National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1979

CONTENTS

	Page
SUMMARY	1
INTRODUCTION	1
OVERVIEW	2
Method of Analysis	2
Blade geometric model	2
Numerical model	3
General Program Description	3
DETAILED PROGRAM PROCEDURE	5
Block Data NGASDAT	6
Main Program NTTACT	6
Subroutine FLOWS	7
Subroutine FLSPLT	7
Subroutine GASTBL	8
Subroutine GAUSS	8
Subroutine GETIN	8
Function HCFRCD	9
Subroutine HCOOL	9
Subroutine HCPINS	9
Subroutine INPRT	9
Subroutine PARRAY	9
Subroutine PLNUM	10
Subroutine PLOTMF	10
Subroutine PREP	10
Subroutine TARRAY	10
Subroutine TCOEF	11
Subroutine THRCON	11
Subroutine WROUT	11
DICTIONARY	11
PROGRAM LISTING	53
REFERENCES	150

SUMMARY

A FORTRAN IV computer program to calculate transient and steady-state temperatures, pressures, and flows in a cooled turbine blade or vane with an impingement insert has been developed and is described in this report. Coolant-side heat-transfer coefficients are calculated internally in the program, with the user specifying one of three modes of heat transfer at each station: (1) impingement, including the effect of crossflow; (2) forced-convection channel flow; or (3) forced convection over pin fins. Additionally, a limited capability to handle film cooling is available in the program. It is assumed that spent impingement air flows in a chordwise direction and is discharged through a split or drilled trailing edge and through film-cooling holes. The program does not allow for radial flow of the spent impingement air. The use of film cooling is restricted by a numerical model requirement for a continuous coolant-channel flow.

Input to the program includes a description of the blade geometry, coolant-supply conditions, outside thermal boundary conditions, and wheel speed. The user can divide the blade by chordwise cuts into as many as 15 slices and can divide each slice into as many as 79 stations around the blade. Each station in turn consists of four calculational nodes through the wall and one in the coolant channel. The blade wall can be two layers of different materials, such as a ceramic thermal-barrier coating over a metallic substrate. Program output includes the temperature at each node, the coolant pressures and flow rates, and the coolant-side heat-transfer coefficients.

INTRODUCTION

As core turbine-engine operating conditions become more severe, it becomes more difficult to effectively cool blades and vanes. Advanced transient thermal calculational techniques are needed to design reliable turbine blades. However, there appears to be no computer program generally available that uses these advanced techniques in combining the required heat-transfer and coolant-flow-distribution calculations. Thus, it was decided to create a computer program that would perform both transient and steady-state heat-transfer and coolant-flow analyses for a cooled blade, given the outside hot-gas boundary conditions, the coolant inlet pressure or flow rate, the geometry of the blade shell, and the cooling configuration.

The resulting program, TACT1, can handle a turbine blade or vane that is equipped with a central coolant-plenum insert from which coolant air flows through holes to impinge on the inner surface of the blade shell or directly into the trailing-edge region. It

is assumed that the spent impingement air then flows chordwise and is dumped through a split or drilled trailing edge and through film-cooling holes. The blade is modeled by dividing it by chordwise planes into as many as 15 slices, with each slice having as many as 79 calculational stations around the blade. Temperatures at each station are calculated for four points through the wall and one in the coolant channel. Included in this model is the capability to analyze a blade with a ceramic thermal-barrier coating. The ability of the program to model film cooling is limited by the numerical flow analysis requirement for a continuous coolant-channel flow.

The TACT1 program is used at the NASA Lewis Research Center on an IBM TSS/360-67 computer. The source program consists of approximately 6000 lines of code and the program requires about 60 000 words of storage. Typical running times for the program are 1.4 seconds of central processor unit (CPU) time per calculational station for a steady-state run and 0.4 second of CPU time per station per time step for a transient run.

The TACT1 program is reported in two parts. This report, part II, is a programmers manual and includes a complete program listing and a detailed description of the procedure. Part I (ref. 1) is a users manual and contains all the information necessary to run the program: a detailed description of the input, the method of solution, and the output as well as a sample problem.

OVERVIEW

Method of Analysis

The details of the analytical method are presented in part I (ref. 1). The blade model used in the analysis is described briefly in this section.

Blade geometric model. - The key to creating a usable computer program is to have as simple a geometric model as possible for the system being analyzed. In this program, the emphasis is on a blade or vane with a central coolant plenum and chordwise flow of the coolant after impingement. Therefore, it was decided that the primary calculational direction would also be chordwise. The blade is divided into layers that are bounded by chordwise cuts through the blade, as shown in figure 1. Each slice is treated separately in the program, with radial heat conduction in the wall the only communication between layers.

Figure 2 gives the details of the geometric model for a single blade slice and shows the breakdown of the blade or vane into calculational stations and nodes. Each calculational station consists of five nodes: one at the wall outer surface, one at the interface between the coating and blade metal, one at a point midway through the wall metal, one at the wall inner surface, and one in the middle of the coolant channel.

For input to the program, the following basic elements of the geometry are needed for each station: (1) the thicknesses of the wall coating and wall metal and the coolant-channel width, (2) the distance of each node from the adjacent lower-numbered node, and (3) the radial span for this slice. In addition, depending on the mode of heat transfer specified, the user must supply impingement-hole diameter and spacing or pin-fin diameter and spacing. Thermal properties of the blade materials must also be specified. The input is described in detail in reference 1.

Numerical model. - The numerical solution for the temperatures throughout the blade involves writing a transient energy-balance equation for each node and forming a set of equations to be solved for the temperature distribution. Similarly, the coolant pressure distribution is determined by writing the transient momentum equation for flow between adjacent fluid nodes and solving the resulting set of equations for static pressures.

The nodal energy balances are linearized, one-dimensional heat-conduction equations at the wall outer-surface node, at the coating-metal interface, and at the wall inner-surface node. At the midmetal node, a linearized, three-dimensional, heat-conduction equation is used. In the coolant channel, energy and momentum equations for one-dimensional compressible flow including friction and heat transfer are written for the elemental channel length between two coolant nodes. The equations used are presented in reference 1.

General Program Description

The TACT1 program is capable of performing a transient analysis as well as a steady-state analysis. In the case of a transient, the program first performs steady-state calculations to determine the initial conditions for the transient.

Figure 3 shows a schematic of the TACT1 calculational scheme. There are three basic, nested calculational loops that must converge for a steady-state solution to be reached. These loops are labeled A, B, and C in figure 3. The program begins a steady-state analysis with the coolant-supply pressure and total coolant flow fixed. The impingement flow is initially assumed to split uniformly at the leading-edge stagnation station, station 1. All coolant flows for the slice under consideration are calculated first, based on the current pressure distribution. The temperatures at each node are then calculated by solving simultaneously the energy equations presented in reference 1. The pressures at each coolant node are calculated by solving simultaneously the momentum equations presented in reference 1. This cycle, loop A in figure 3, is repeated until the pressure distribution no longer changes. The flow split between suction- and pressure-side coolant channels is then checked by comparing the pressures at the ends of the two channels. If they do not match, the impingement flow split at the leading edge

is adjusted and the inner loop calculations are repeated. This adjustment comprises loop B in figure 3. Once the proper flow split is achieved, the program moves up the blade to the next slice and repeats this sequence. After all N slices have converged, the total coolant mass flow used is compared with the inlet coolant mass flow. If there is an imbalance, either the inlet flow or the supply pressure is adjusted, depending on which was specified in the input; and the calculations start over. This is loop C in figure 3. Once the overall coolant mass flow balance is satisfied, the steady-state solution is complete and the transient calculations begin. During a transient calculation, loop B is bypassed because the coolant flow-split is primarily a function of blade geometry. Loop C is also bypassed because the inlet coolant mass flow rate at a given time is estimated based on the coolant mass flow used at the previous time step and the change in supply pressure.

The TACT1 subprograms and the calling relations are shown in figure 4. Block data subprogram NGASDAT contains air properties, for use in TACT1, tabulated as functions of temperature at a pressure of 20 atmospheres from reference 2. This subprogram must be loaded before execution of the program. The main program, NTTACT, calls other subroutines in their proper order.

The first call from NTTACT is to GETIN, a subroutine that controls the reading, storing, and printing of input data. Subroutine GETIN calls INPUT to print the input data if the user specifies INEDIT > 0. Subroutine INPRT has a call to PREP to put the input data in its proper form for use. All data are input by using a NAMELIST format.

After the input data have been read, the number of time steps, NTYM, to be used in the transient is determined in NTTACT. If only a steady-state solution is to be calculated, NTYM = 1. Time-dependent boundary conditions are then evaluated, with the initial entries assumed to be steady-state values. Then NTTACT loops through the blade, calling on subroutines PLNUM, PREP, and TCOEF for each slice. The first time through is a steady-state calculation.

Subroutine PLNUM calculates the pressure distribution in the impingement plenum for the current slice, given the inlet pressure and coolant flow-rate. PLNUM calls GASTBL for gas properties.

Subroutine PREP extracts the input data for the current slice from the input arrays.

Subroutine TCOEF controls loop A in figure 3, the iterative calculations of temperature and pressure for the nodes of the current slice. Each iteration in TCOEF requires calls to subroutines FLOWS, HCOOL, THRCON, TARRAY, PARRAY, and GAUSS.

Subroutine FLOWS computes the impingement jet flow rates, coolant-channel mass flow rates, and channel Mach numbers for each station around the blade, given the plenum pressure and temperature and the current pressure distribution in the coolant channel. FLOWS calls GASTBL for gas properties.

Subroutine HCOOL is called to calculate coolant heat-transfer coefficients for all the stations of this slice, based on the latest values of mass flow rate. HCOOL calls function HCFRCD to calculate forced-convection heat-transfer coefficients and GASTBL for gas properties.

Subroutine THRCON determines the wall thermal conductivity from the input table of conductivity as a function of temperature.

Subroutine TARRAY sets up the array of coefficients for the conduction and convection equations for each node. Calls are made to HCPINS for pin-fin heat-transfer coefficients, to HCFRCD for forced-convection heat-transfer coefficients, and to GASTBL for gas properties. TCOEF calls subroutine GAUSS to solve the set of equations for the temperature at each node.

Subroutine PARRAY sets up the array of coefficients for the momentum equations in the coolant channels and TCOEF calls subroutine GAUSS to solve the set of equations for the pressure at each coolant node.

After a new set of temperatures and pressures has been determined, convergence is checked by using the coolant-channel pressure at the blade leading edge. If this pressure stays within a tolerance band for four successive iterations, convergence is accepted. Once convergence is achieved, TCOEF calls subroutine FLSPLT to check the coolant flow-split between the pressure and suction sides. This is loop B in figure 3. Initially, the impingement jet flow at the forward stagnation station is assumed to split evenly between the suction- and pressure-side channels. If the coolant-channel pressures at the end of the impingement insert do not match, the flow split at the forward stagnation station is adjusted to increase the flow to the channel with the higher pressure at the end of the insert, and iteration loop A is repeated. Once a satisfactory flow split has been achieved, TCOEF calls subroutine WROUT to print the output for this slice and calls subroutine PLOTMF if there is to be graphical output. After NTTACT has calculated all blade slices, the total coolant mass flow is compared with the impingement-plenum inlet mass flow rate used to start the calculations. If the two flow rates are not close enough, the inlet mass flow or supply pressure is adjusted and the calculations are repeated. This is loop C in figure 3.

When the initial steady-state solution has been completed, the transient calculations are started. The transient is continued until the time reaches the specified maximum.

Subroutine PLOTMF makes use of a TSS/360 graphics package at the NASA Lewis Research Center to plot temperature and pressure distributions for the blade.

DETAILED PROGRAM PROCEDURE

Table I lists the names of each of the subprograms in TACT1, the corresponding

TSS/360 source module names, the COMMON blocks used in each, the names of the subroutines called by each, and the names of subroutines calling each. Table II is a cross-reference listing of named COMMON blocks and the subprograms using them. This section gives a detailed description of each subprogram used in TACT1. All variable names used are defined in the section DICTIONARY. The BLOCK DATA subprogram and the MAIN PROGRAM are discussed first and then each subprogram is described, in alphabetical order.

Block Data NGASDAT

A BLOCK DATA subprogram, NGASDAT, is used to provide a table of gas properties to the program. The properties are put in the array GS through a DATA statement with 5×NG entries, where NG is the number of table entries for each property. The first NG values are temperatures, the second are thermal conductivities, the third are specific heats, the fourth are Prandtl numbers, and the final NG values are viscosities. The property values included are taken from reference 2 at a pressure of 20 atmospheres.

Main Program NTTACT

The MAIN PROGRAM for TACT1, NTTACT, has overall control of the program. Figure 5 is a flow chart for NTTACT. During initialization, a call is made to a system subroutine, TIME, to get a unique label to be used to identify the plotted output for a given run. After the call to GETIN, where all the input data are read, NTTACT initiates the solution procedure by searching the transient boundary condition tables and using linear interpolation to extract the values for the current time. The next step is to begin the loop, labeled C in figure 3. The solution progresses from hub to tip. For each slice, NTTACT calls PLNUM to calculate coolant-supply conditions; PREP to extract the input data from the input tables; and TCOEF to calculate flows, temperatures, and pressures. After the return from TCOEF, NTTACT updates the total amount of coolant used, WUSED, by adding the amount used in the current slice, WIM. The amount of coolant-plenum flow available for the remaining slices, WPLEN, is updated by subtracting WIM. After all slices have been done, the overall amount of coolant used is printed and then checked against the assumed coolant flow-rate. If the absolute value of the difference, EXCESW, is more than 1 percent of the assumed flow, the assumed flow or the supply pressure is adjusted and the calculations are repeated. For transient runs, after the initial steady-state coolant-flow balance, there are no more iterations on coolant flow. Instead the flow for a given time step is based on the actual flow used in

the preceding time step and on the ratio of supply pressure for the two steps. Finally, once all loops have been completed, NTTACT calls PLOTMF to get a final summary plot of blade temperatures.

Subroutine FLOWS

Subroutine FLOWS is a routine to calculate the flow rates through all impingement and film-cooling holes, the friction factor in the coolant channels, and film-cooling effectiveness. FLOWS makes use of the current impingement-plenum mean pressure and temperature and coolant-channel pressure and temperature distributions. The impingement jet flow-rate, WJ, is calculated for each station in the forward region and checked against the choked flow-rate, WCR. If WJ is greater than WCR, then WJ is set equal to WCR. If there is any film cooling on the blade, the film-cooling flow rates in the forward region, WFC, are also calculated. Then, the coolant-channel flow rates, WCROS, are computed by considering a mass balance between stations, as illustrated in figure 6. Once the forward-region coolant flows have been determined, the Reynolds numbers - RE for the coolant channel, and REFC for the film-cooling flow - and the square of the coolant Mach number, AM2, are computed for each forward station.

The next step is to calculate the amount of coolant, WDUMP, dumped directly into the trailing-edge region from the coolant plenum. Then the total amount of coolant used for this slice, WIM, is determined by summing the impingement jet flows and WDUMP. Following this, the flows in the trailing-edge region are computed, with the coolant flow being reduced by the amount of any film-cooling flow. Then, trailing-edge-region values of RE, REFC, and AM2 are calculated.

After all the coolant flow-rates are determined, the friction factor, FF, is calculated at each station. Finally, if there is any film cooling used, the film effectiveness is calculated by using the method of reference 3.

Subroutine FLSPLT

Subroutine FLSPLT is used to determine the location of the stagnation impingement jet, station JS, and the fraction of that jet's flow that splits to each side of the blade, DELTAN. Figure 7 is a detailed flow chart for subroutine FLSPLT. The primary variable carried into FLSPLT is the pressure-match parameter, EPSN, which is defined as

$$\mathrm{EPSN} = \frac{(\mathrm{P(2, ISLICE, NFWD-1) - P(2, ISLICE, NFWD)})}{\mathrm{P(2, ISLICE, NFWD-1)}} \tag{1}$$

where the pressures are as illustrated in figure 8.

The magnitude and sign of EPSN are used to determine the adjustment of the stagnation impingement-jet row location and the fraction of that jet that splits to the suction-side channel. Initially, the stagnation jet row is located at station 1 and the split is DELTAN = 0.50. If EPSN is positive, DELTAN is set to 0.75 to increase the flow down the suction-side channel; if EPSN is negative, DELTAN is set to 0.25 to increase the flow down the pressure-side channel. For subsequent entries into FLSPLT, the value of DELTAN is adjusted by passing a straight line through the last two points on a plot of EPSN versus DELTAN and picking the value of DELTAN where this line crosses the axis at EPSN = 0. If this intercept falls outside the DELTAN range of 0 to 1, the stagnation station, JS, must be moved to an adjacent station and DELTAN set to 0.50. Once a sign change is observed in EPSN, a fine-tuning process is triggered in FLSPLT. In this case, the values of DELTAN and EPSN for the iteration preceding the sign change are saved and used as one of the points of the straight-line interpolation scheme for all subsequent iterations.

Subroutine GASTBL

Subroutine GASTBL is used to interpolate in the array GS for gas properties, given the absolute temperature. Linear interpolation is used.

Subroutine GAUSS

Subroutine GAUSS is a routine that uses Gaussian elimination to solve a set of simultaneous equations. The array of coefficients, TCOF, is in the form of a compressed, augmented band matrix. That is, only the matrix elements within the band and the constants from the right side are stored in TCOF. The matrix band width, K, is determined by the node-numbering system used. In TACT1, the temperature calculations require a band width of 23 elements, and the pressure calculations require 19.

Subroutine GETIN

Subroutine GETIN is a routine used to initialize input-data default values and to read and store input data. Input is in NAMELIST form as described in reference 1. The entire data set is read, and the input variables for each slice are stored in two arrays: INDCHN for integer data, and CHANL for real-number data. If the input is provided in SI units, subroutine GETIN converts it to U.S. customary units for internal use. If the user specifies INEDIT > 0, GETIN calls subroutine INPRT to print out the input data.

Function HCFRCD

Function subprogram HCFRCD is a routine to calculate a turbulent, forced-convection heat-transfer coefficient for channel flow as described in reference 1.

Subroutine HCOOL

Subroutine HCOOL is a routine containing the correlations for impingement heat transfer. The first part of HCOOL deals with leading-edge-region impingement cooling. In this part, the inner surface length from the stagnation impingement jet to the end of the leading-edge impingement region is determined and then used in a correlation to compute the average heat-transfer coefficient in this region. Beyond this region, for stations starting at ICOR, calculations are done by using an impingement-with-crossflow correlation.

Subroutine HCPINS

Subroutine HCPINS is a routine to calculate coolant-side heat-transfer coefficients in regions of the blade equipped with pin fins. In addition, the effective heat-transfer area, which accounts for the pin surface area and the pin-fin effectiveness, is calculated.

Subroutine INPRT

Subroutine INPRT is a routine to print a listing of the input data. Also, INPRT sets up the initial temperature distribution in the blade. Subroutine PREP is called for each slice to extract input data from the arrays INDCHN and CHANL.

Subroutine PARRAY

Subroutine PARRAY is a routine to set up the matrix to be solved for coolant-channel pressure distribution. The equations used are detailed in reference 1.

The array of coefficients generated in PARRAY, TCOF, is in the form of a compressed, augmented band matrix. Coefficients that would be on the main diagonal of the full matrix are stored in column 10 of the TCOF array. The terms from the right side of the equations are stored in column 20 of TCOF.

Subroutine PLNUM

Subroutine PLNUM is a routine to calculate the pressure and temperature distributions in the central coolant plenum. The mean plenum static pressure and temperature for each slice are used as the supply conditions for the impingement jets. The total temperature and pressure at the outlet of one plenum slice are used as input for the next slice.

There are five arguments used in the call statement for PLNUM: WXX is the mass flow rate into this plenum slice; PXX and TXX are the calculated, average static pressure and temperature; and PTEXIT and TTEXIT are total temperature and pressure, respectively. Going into the subroutine, PTEXIT and TTEXIT are the values at the entrance to this slice. On return, they are the values at the exit of this slice.

Subroutine PLOTMF

Subroutine PLOTMF is a routine that plots TACT1 output. PLOTMF makes use of a TSS/360 graphics package at the NASA Lewis Research Center. For an installation without this specific package, this subroutine would have to be revised or bypassed.

PLOTMF plots temperature and pressure versus surface distance from station 1 for each slice of the blade for a steady-state case. For transients, a set of two summary plots is made for each time step: the plots contain temperatures for all slices on one graph.

Subroutine PREP

Subroutine PREP is a routine to extract input data from storage and put it in the form used in the calculations. In PREP, the hot-gas-side boundary condition tables are searched and linear interpolation is done to extract the boundary condition values at each calculation station at the given time.

Subroutine TARRAY

Subroutine TARRAY is a routine to set up the matrix to be solved for the temperatures in each slice. The equations used are detailed in reference 1.

The array of coefficients generated in TARRAY, TCOF, is in the form of a compressed, augmented band matrix. The 12th column of TCOF contains the elements that would be on the main diagonal of a full matrix. The terms from the right side of the equations are stored in column 24.

Subroutine TCOEF

Subroutine TCOEF is a routine that controls the calculations for flow rates, temperatures, and pressures. The first time TCOEF is entered for each slice an initial estimate of the coolant-channel pressure distribution is set up. TCOEF controls the iterations in loops A and B in figure 3. Loop A consists of calls to subroutines FLOWS, TARRAY, and PARRAY. The variable IVERGE is used to count the number of iterations in loop A. Convergence is checked by comparing the four most-recent values of coolant-channel pressure at the flow-split point. When the ratio of the maximum difference among these four to the difference between coolant-supply pressure and trailing-edge exit pressure is less than PCNVRG, loop A is complete. Then the flow split at the stagnation impingement jet, JS, is checked by subroutine FLSPLT and adjusted if necessary. Loop B involves repeating loop A for a new flow split. The variable IDELT is used to count the number of flow-split iterations in loop B. Once flow-split convergence is achieved, WROUT is called to print the output for the current slice.

Subroutine THRCON

Subroutine THRCON is a routine that takes the wall temperatures and searches for the thermal conductivity values in the input tables.

Subroutine WROUT

Subroutine WROUT is a routine to control the printing of the output from TACT1. Output units are the same as the input data units.

DICTIONARY

All the important variable names used in the TACT1 code are defined in this section. The only names not defined are locally used indices. All dimensioned variables include the dimensions. The dictionary also indicates the COMMON block or subroutine in which each variable is used.

Variable	Common	Subroutine	Definition
A(400)	TCO		cross-sectional area normal to chordwise direction, in^2, accessed by node number
AA		GETIN	outer-surface length between stations, in., used for calculating interpolated values of TDLX(2), TDLX(3), and TDLX(5)
AA		PLNUM	coolant-plenum cross-sectional area, in^2, used in plenum pressure-drop calculations
AB		PLNUM	maximum Mach number in coolant plenum
AC(5)		GASTBL	array of interpolated values of gas properties
ACH		PLNUM	coolant-plenum choked-flow indicator
ADUMP	TCO		area of slot or jets dumping coolant directly into trailing-edge region, in^2
ADUMPC		INPRT	same as ADUMP, but converted to cm^2 for input listing when input is in SI units
AHG		PREP	intermediate value of hot-gas-side heat-transfer coefficient, $Btu/hr \cdot ft^2 \cdot {}^oF$, used for interpolating in input table during a transient
AHTRN1		TARRAY	inner-surface area for heat-transfer purposes, in^2
AHTTR		HCPINS	total surface area in pin-fin channel, in^2
AINTRV		PLOTMF	floating-point form of number of temperature intervals in summary plots
AJ		PLNUM	floating-point form of indicator J - 1
AJET(80)	TCO		total area of impingement jets at each station, in^2

Variable	Common	Subroutine	Definition
AJET(80)		FLSPLT	total area of impingement jets at each station, in^2, carried into subroutine as argument
AKC(15, 80)	TCO		wall outer-coating thermal conductivity, Btu/hr • ft • ^{o}F
AKCTBL(20)	BOUND		input table of wall outer-coating thermal conductivity, Btu/hr • ft • ^{o}F, versus temperature, ^{o}F
AKW(15, 80)	TCO		wall metal thermal conductivity, Btu/hr • ft • ^{o}F
AKWTBL(20)	BOUND		input table of wall metal thermal conductivity, Btu/hr • ft • ^{o}F, versus temperature, ^{o}F
ALABL(7)		PLOTMF	array containing time and date label for identification of output plots
ALPH(12)		NTTACT	alphameric array used to uniquely identify output of each job
ALPHA	FRIC		constant used in friction factor calculations
ALPH2(4)		PLOTMF TCOEF NTTACT	time and date information, generated in NTTACT and passed to plotting subroutine as argument
AM		HCOOL	exponent on Reynolds number in Kercher-Tabakoff impingement correlation
AMC(20)		PLNUM	Mach number distribution in coolant plenum for a given slice
AMCHOK		FLOWS	if any stations have a Mach number greater than 1.0, the value is saved in this variable and returned as an argument, to be printed by TCOEF
AMIN		FLOWS HCPINS	area of coolant-flow channel at a given station, reduced by pin-fin blockage

Variable	Common	Subroutine	Definition
AM2(80)	TCO		array containing square of coolant-channel Mach number at each station, for a given slice
APG		PREP	intermediate value of hot-gas-side pressure, lbf/in^2, used for interpolating in input table during transient
APLEN		GETIN	input value of coolant-plenum area for given slice, cm^2 (in^2)
APLN(15)	RADL		internal array to store plenum area for each slice, in^2
AP1		GASTBL	interpolating factor in gas property table lookup
AP2		GASTBL	1.0 - AP1
AQG		PREP	intermediate value of hot-gas-side heat flux, $Btu/hr \cdot ft^2$, used for interpolating in input table during transient
ASTG		TCOEF	inner-surface area under stagnation-point impingement jet, in^2
ATG		PREP	intermediate value of hot-gas-side temperature, ^{o}R, used for interpolating in input table during transient
ATMAXP		PLOTMF	adjusted maximum temperature, ^{o}F, used as high endpoint on output plots
ATMINP		PLOTMF	adjusted minimum temperature, ^{o}F, used as low endpoint on output plots
ATYME		PLOTMF	value of time in transient, sec, used on output plots for identification
AVRGA		PARRAY	area ratio used in momentum equation at entrance to trailing edge
AZ		PLNUM	dummy variable, used as either diameter-area ratio or flow adjustment

Variable	Common	Subroutine	Definition
A1		FLOWS	interpolating factor in friction factor calculation in transitional Reynolds number range
A1		TARRAY	upstream half of inside-wall heat-transfer area, in^2, associated with coolant-channel node
A2		FLOWS	interpolating factor in friction factor calculation in transitional Reynolds number range
A2		TARRAY	downstream half of inner surface heat-transfer area, in^2, associated with coolant-channel node
A3		TARRAY	same as A1, but on opposite wall, only used in trailing-edge region
A4		TARRAY	same as A2, but on opposite wall, only used in trailing-edge region
B		GETIN	ratio of length to thickness, used along with AA to calculate interpolated values of TDLX(2), TDLX(3), and TDLX(5)
B(20)		PLNUM	spanwise static-temperature distribution, ^{o}R, in coolant plenum for given slice
BC		GETIN	NAMELIST name
BCHGP(1000)	BOUND		input table of hot-gas, pressure-side heat-transfer coefficients, $W/m^2 \cdot K$ ($Btu/hr \cdot ft^2 \cdot {}^{o}F$)
BCHGS(1000)	BOUND		input table of hot-gas, suction-side heat-transfer coefficients, $W/m^2 \cdot K$ ($Btu/hr \cdot ft^2 \cdot {}^{o}F$)
BCPGP(1000)	BOUND		input table of hot-gas, pressure-side relative static pressure, kPa (lbf/in^2)
BCPGS(1000)	BOUND		input table of hot-gas, suction-side relative static pressure, kPa (lbf/in^2)

Variable	Common	Subroutine	Definition
BCQGP(1000)	BOUND		input table of hot-gas, pressure-side heat flux, W/m^2 ($Btu/hr \cdot ft^2$)
BCQGS(1000)	BOUND		input table of hot-gas, suction-side heat flux, W/m^2 ($Btu/hr \cdot ft^2$)
BCTGP(1000)	BOUND		input table of hot-gas, pressure-side temperature, K (^oF)
BCTGS(1000)	BOUND		input table of hot-gas, suction-side temperature, K (^oF)
BCTIME(50)	BOUND		input table of time at which transient input tables are specified, sec
BCXP(400)	BOUND		input table of outer-surface, pressure-side locations at which hot-gas conditions are input, cm (in.)
BCXS(400)	BOUND		input table of outer-surface, suction-side locations at which hot-gas conditions are input, cm (in.)
BES		HCOOL	equivalent slot width, in., used in leading-edge impingement correlation
BETA	FRIC		constant used in friction factor calculations
BETA1		PLNUM	square of pressure at inlet to coolant plenum for given slice, $(lbf/in^2)^2$
BETTA(20)		PLNUM	spanwise static-pressure distribution in coolant plenum, lbf/in^2
BTA	TCO		indicates type of hot-gas boundary condition
C		FLOWS GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	gas thermal conductivity, $Btu/hr \cdot ft \cdot {}^oF$

Variable	Common	Subroutine	Definition
CD	TCO		impingement-jet discharge coefficient
CDEN(2)	UNITS		conversion factor for density units
CD1(200)		NTTACT	dummy variable used to print selected intermediate temperature values
CEXCSW		NTTACT	amount of excess coolant flow, in SI units, kg/hr
CGASC(2)	UNITS		conversion factor for gas constant
CH(15)		PLNUM	coolant-channel choking indicator
CHANL(8000)	SPECL		array for storing input data
CHANLS		GETIN	NAMELIST name
CHFLX(2)	UNITS		conversion factor for heat-flux units
CHTC(2)	UNITS		conversion factor for heat-transfer-coefficient units
CIMP1 CIMP2 CIMP3 CIMP4 CIMP5 CIMP6 CIMP7	IMPCOR		user-supplied constants for general impingement correlation
CINCH(2)	UNITS		conversion factor for length units
CMSFL(2)	UNITS		conversion factor for mass flow rate units
CNUM(80)	TCO		number of impingement jets at each station for given slice
CONDCT		HCOOL	coolant-air thermal conductivity, Btu/hr · ft · ^{o}F
CONTRL		GETIN	NAMELIST name
CP	TCO		gas specific heat at constant pressure, Btu/lbm · ^{o}F

Variable	Common	Subroutine	Definition
CPC(80)	PRPS		coolant specific heat at constant pressure at each coolant node for given slice, Btu/lbm · ^{o}F evaluated at a mean temperature between bulk coolant temperature and wall temperature
CPIM		NTTACT PLOTMF	mean impingement-plenum pressure for given slice, in SI units, kPa
CPM		FLOWS	hot-gas-stream specific heat at constant pressure, Btu/lbm · ^{o}F
CPO	PRPS		specific heat at constant pressure, Btu/lbm · ^{o}F, evaluated at impingement-jet supply temperature
CPRSR(2)	UNITS		conversion factor for pressure units
CRHOVG(2)	UNITS		conversion factor for density × velocity units
CRITR		FLSPLT	coolant flow-split convergence criterion
CSPHT(2)	UNITS		conversion factor for specific-heat units
CTCON(2)	UNITS		conversion factor for thermal conductivity units
CTMPF(2)	UNITS		conversion factor for temperature units
CT0G		NTTACT	mean impingement-plenum static temperature for given slice, in SI units, K
CURV		TARRAY	factor to account for wall curvature in heat-conduction equations
CVISC(2)	UNITS		conversion factor for viscosity units
CWPLEN		NTTACT	coolant-plenum flow rate at entrance to given slice, in SI units, kg/hr
CWUSED		NTTACT	total amount of coolant air used, in SI units, kg/hr
CX		PLNUM	function of isentropic exponent k, $-(k+1)/[2(k-1)]$

Variable	Common	Subroutine	Definition
C1		PLNUM	function of isentropic exponent k, 2k/(k - 1)
C3		PLNUM	computed constant involving wheel speed and isentropic exponent
C3		FLOWS	ratio of specific heats at constant pressure, coolant to hot gas
C5		PLNUM	computed constant involving isentropic exponent and gas constant
C6		PLNUM	function of isentropic exponent k, (k - 1)/2
C7		PLNUM	computed constant involving isentropic exponent and gas constant
C8		PLNUM	computed constant involving isentropic exponent and gas constant
D		PLNUM	convergence parameter in coolant-plenum pressure calculations
DD		PLNUM	coolant-plenum hydraulic diameter, in
DEH		HCOOL	hydraulic diameter of equivalent slot, in., used in leading-edge impingement correlation
DELAST		FLSPLT	variable used to save flow-split fraction at which pressure-match parameter, EPSN, changes sign
DELTA	FRIC		constant used in friction factor calculation
DELTAN(15)		FLOWS FLSPLT HCPINS TARRAY TCOEF WROUT	fraction of stagnation-point impingement-jet flow that splits to suction-side coolant-flow channel for each slice
DELTAO		FLSPLT	value of DELTAN from previous flow-split iteration

Variable	Common	Subroutine	Definition
DENOM		NTTACT	intermediate variable used in time interpolation of some boundary conditions
DH(80)	TCO		coolant-channel hydraulic diameter at each station, in.
DHF(80)	TCO		effective diameter of film-cooling hole at each station, in., defined as hydraulic diameter of one hole multiplied by square root of number of holes at station
DHJ(80)	TCO		actual hydraulic diameter of an impingement hole at each station, in.
DHYD		GETIN	input value of coolant-plenum hydraulic diameter for a slice, cm (in.)
DIFN		TCOEF	pressure difference parameter used in checking convergence
DIFO		TCOEF	maximum pressure difference parameter used in checking convergence
DIFTOL		PLNUM	tolerance on coolant-plenum pressure-drop calculations
DIMP1 DIMP2 DIMP3 DIMP4 DIMP5 DIMP6	IMPCOR		user-supplied constants for leading-edge impingement correlation
DLTAOP		FLSPLT	best value of DELTAN in the event of an unstable flow split
DLTYME	TRNSNT		time step used in transient calculations, sec
DLX(400)	TCO		chordwise distance from each node to adjacent upstream node, in.
DP(80)	PRPS		diameter of pin fins at each station, in.

Variable	Common	Subroutine	Definition
DPLN(15)	RADL		coolant-plenum hydraulic diameter for each slice, in.
DR		PLNUM	radial length increment in coolant-plenum calculations, in.
DR2		PLNUM	radial increment squared, in^2
DUMR1(80)	PRPS		dummy variable, not currently used
DUMR2(80)	PRPS		dummary variable, used to carry impingement-jet Reynolds number to output subroutine
DUMTER		PARRAY	intermediate variable in momentum equation involving coolant dumped directly into trailing edge
DUM1(10)		INPRT WROUT	dummy variables used to print input listings and program output
DUM2(10)		INPRT WROUT	
DUM3(10)		INPRT	
DUM4(10)			
DUM5(10)			
DUM6(10)			
DUM7(10)			
DUM8(10)			
DUM9(10)			
DUM10(10)			
DUM11(10)			
DUM12(10)			
DUM13(10)			
DUM14(10)			
DUM15(10)			
DUM16(10)			

Variable	Common	Subroutine	Definition
DUM17(10)		INPRT	dummy variables used to print input listings and program output
DUM18(10)			
DUM19(10)			
DUM20(10)			
DUM25(10)			
DUM52(10)			
DUM53(10)			
DUM55(10)			
DX		PLNUM	spanwise step size used in calculating coolant-plenum pressure and temperature distributions
DXTEMP		PLNUM	variable to temporarily hold DX
DX1		TARRAY	path length between midwall node and adjacent upstream midwall node, in.
DX10		TARRAY	path length between outer coating - wall junction node and adjacent downstream outer coating - wall junction node, in.
DX2		TARRAY	path length between midwall node and adjacent downstream midwall node, in.
DX3		TARRAY	path length between outer-surface node and adjacent upstream outer-surface node, in.
DX4		TARRAY	path length between outer-surface node and adjacent downstream outer-surface node, in.
DX5		TARRAY	path length between inner-surface node and adjacent upstream inner-surface node, in.
DX6		TARRAY	path length between inner-surface node and adjacent downstream inner-surface node, in.

Variable	Common	Subroutine	Definition
DX7		TARRAY	path length between coolant node and adjacent upstream coolant node, in.
DX9		TARRAY	path length between outer coating - wall junction node and adjacent upstream outer coating - wall junction node, in.
D1		PLNUM	computed constant used in coolant-plenum pressure equations to account for effect of pumping due to wheel rotation
D2		PLNUM	computed constant used in coolant-plenum temperature equations to account for effect of pumping due to wheel rotation
E		PLNUM	factor used in adjusting convergence rate in coolant-plenum calculations
EFAREA(80)		HCPINS TARRAY	effective area, in^2, for heat transfer at stations with pin fins, including pin-fin effectiveness for heat transfer
EFTVNS		HCPINS	pin-fin effectiveness
EMES(80)	FLMCOL		for film cooling, ratio of coolant mass flux to free-stream mass flux, multiplied by equivalent slot width
EML		HCPINS	term used in pin-fin effectiveness calculation
ENDEFF		TARRAY	term in heat-transfer equations to account for convection to rear edge of blade when heat-transfer coefficients are input
ENDFLX		TARRAY	term in heat-transfer equations to account for convection to rear edge of blade when heat flux is input
EPLAST		FLSPLT	variable used to save latest value of pressure-match parameter, EPSN

Variable	Common	Subroutine	Definition
EPS	FRIC		constant used in friction factor calculations
EPSMIN		FLSPLT	minimum value attained by pressure-match parameter, EPSN, for unstable flow split
EPSN		FLSPLT TCOEF	pressure-match parameter, defined as difference between suction- and pressure-side coolant-channel static pressures at end of insert, divided by suction-side coolant-channel static pressure
EPSO		FLSPLT	old value of pressure-match parameter, EPSN
ETAPRM		FLOWS	film-cooling effectiveness based on ratio of enthalpy differences
EXCESW		NTTACT	amount of excess coolant flow, difference between inlet flow and that actually used, lbm/hr
FACTOR		TARRAY	special variable to adjust amount of energy carried in by an impingement jet for case of a calculation station adjacent to flow-split station
FF(80)	TCO		value of friction factor at each flow station
FILM		PARRAY	term to account for momentum carried off by film-cooling air
FILMW		TARRAY	total film-cooling flow from given coolant node
FLMEFF(80)	FLMCOL		film-cooling effectiveness based on ratio of temperature differences
FM		GAUSS	multiplying factor used in Gauss elimination scheme

Variable	Common	Subroutine	Definition
FUNP		PLNUM	statement function to calculate pressure difference in coolant plenum
FUNT		PLNUM	statement function to calculate temperature difference in ccolant plenum
F1(20)		PLNUM	friction factor in coolant plenum
GAM	TCO		ratio of specific heats
GAMC(80)	PRPS		ratio of specific heats at each coolant-channel node
GAMO	PRPS		ratio of specific heats at coolant-supply conditions
GEO		GETIN	NAMELIST name
GG		HCOOL	mass flux ratio, coolant crossflow to impingement-jet flow
GI		HCOOL	momentum flux ratio, coolant crossflow to impingement-jet flow
GMASS		HCOOL	mass flux from row of leading-edge impingement holes
GS(200)	GAAS		table of gas properties
G1		PLNUM	computed constant in coolant-plenum calculations, involving flow rate, gas constant, and specific heat at constant pressure
G2		PLNUM	computed constant in coolant-plenum calculations, involving flow rate and gas constant
HBAR		WROUT	average coolant-side heat-transfer coefficient for given slice, $W/m^2 \cdot K$ ($Btu/hr \cdot ft^2 \cdot {}^oF$)
HC(80)	TCO		coolant-side heat-transfer coefficients at each station for given slice, $Btu/hr \cdot ft^2 \cdot {}^oF$

Variable	Common	Subroutine	Definition
HCAL (4)		INPRT	alphameric array containing labels identifying type of coolant-side heat transfer
HG (80)	TCO		hot-gas-side heat-transfer coefficient at each station for given slice, Btu/hr · ft^2 · ^{o}F
HSTGMX		TCOEF	maximum physically possible value of coolant heat-transfer coefficient under stagnation jet, Btu/hr · ft^2 · ^{o}F
HUB1		TARRAY	term in conduction equation to account for specified hub temperature
HUB3		TARRAY	term in conduction equation to account for specified hub heat flux
HX		TARRAY	multiplying factor on coolant heat-transfer coefficient, initialized to 1.0 but may be changed dynamically
HYCOS		HCPINS	hyperbolic cosine term
HYSIN		HCPINS	hyperbolic sine term
IADJIN	SPECL		input variable that indicates which coolant-plenum supply property is to be held fixed
ICHK		PARRAY	indicates which side of blade a given station is on: 0 if suction side, 1 if pressure side
ICHNL		INPRT PREP	slice number, carried through as argument
ICHOKE		FLOWS PARRAY TCOEF	number of station that shows choked coolant flow
ICOMP		TARRAY	number of station adjacent to impingement flow-split station in pressure-side direction

Variable	Common	Subroutine	Definition
ICOMS		TARRAY	number of station adjacent to impingement flow-split station in suction-side direction
ICONV		FLSPLT TCOEF	indicator for convergence of flow-split iterations
ICOR	TCO		station at which use of impingement-with-crossflow correlation is to begin
IDELT		FLSPLT TCOEF WROUT	counter of number of flow-split iterations performed
IDN		PARRAY	downstream node number for coolant-channel pressure calculations
IDNS		PARRAY	downstream station number for coolant-channel pressure calculations
IDX		PARRAY	upstream node number for coolant-channel pressure calculations
IEND		GETIN	last point in CHANL array occupied by data for given slice
IFCP		FLOWS	indicator used in locating pressure-side film-cooling holes
IFCS		FLOWS	indicator used in locating suction-side film-cooling holes
IFILM	TCO		input indicator for film cooling
IFLU		PREP	coolant-channel node number
IFNL		PARRAY	number of coolant-channel nodes
IFNL		TCOEF	total number of stations, minus 1
IFSPLT		FLOWS	indicates in which direction film-cooling air flows from stagnation station
IGG(80)		HCOOL	array containing node numbers at which ratio of coolant crossflow to impingement-jet flow is out of Kercher-Tabakoff correlation range

Variable	Common	Subroutine	Definition
IGGC		HCOOL	counts number of entries in IGG array
IHC(80)	TCO		indicates type of coolant-side heat transfer at each station for given slice
IHCT		GETIN	input value of IHC for given station
IHUB	TCO		indicates type of boundary to be used at hub end of blade
II		HCOOL	coolant-channel node number
IIHCTZ		GETIN	locates IHC array in overall array INDCHN
ILEAD		HCOOL	last station in range of leading-edge impingement correlation
IMS		FLOWS	location of film-cooling hole preceding current station
INDCHN(2000)	SPECL		array for storing integer input data
INEDIT		GETIN INPRT	input variable to control listing of input data
INN		PREP	location of IHC data in INDCHN array
INSTRT		GETIN	starting point for storage of integer data in INDCHN array for given slice
INUM		NTTACT	number of stations on each side of blade
IN1		PREP	location of end of group of single-valued variables in INDCHN array
IPLOT	SPECL		input indicator to control plotting options
IPRES		WROUT	pressure-side, outer-surface node number
IPRSMN		WROUT	location of minimum outer-surface temperature on pressure side
IPRSMX		WROUT	location of maximum outer-surface temperature on pressure side

Variable	Common	Subroutine	Definition
IRE(80)		HCOOL	list of station numbers at which impingement-jet Reynolds number is out of Kercher-Tabakoff correlation range
IRL		PARRAY	coolant node number
ISBLOK	TCO		starting point in CHANL array of data for given slice
ISEN		TARRAY	indicates which side of blade station IS is on
ISENS		FLOWS	indicates which side of blade a given station is on
ISENS		TARRAY	indicates which side of blade a given trailing-edge region station is on
ISLICE	TCO		current slice number
ISTA		GETIN	input station number
ISTAT		WROUT	outer-surface node number for given station
ISTATD		WROUT	outer-surface node number immediately downstream of ISTAT
ISTB		GETIN	input station number
ISTRT		HCOOL	station at which use of Kercher-Tabakoff correlation begins
ISTRT		TCOEF	first station in trailing-edge region
ISUCMN		WROUT	location of minimum outer-surface temperature on suction side
ISUCMX		WROUT	location of maximum outer-surface temperature on suction side
ISUCT		WROUT	suction-side station number
ISUP		TARRAY	adjacent station, upstream of current station, IS
ISYM(5)		PLOTMF	data array containing plotting symbol codes

Variable	Common	Subroutine	Definition
ITDHFZ		GETIN	locates film-cooling-hole data in CHANL array for given slice
ITDHJZ		GETIN	locates impingement-hole data in CHANL array for given slice
ITDLXZ		GETIN	locates node spacing data in CHANL array for given slice
ITDPZ		GETIN	locates pin-fin diameter data in CHANL array for given slice
ITHKZ		GETIN	locates wall and channel thickness data in CHANL array for given slice
ITIP	TCO		indicates type of boundary to be used at tip end of blade
ITRBG		INPRT	first station in trailing-edge region
ITRBG		WROUT	first station in trailing-edge region
ITREO		WROUT	last outside node at trailing edge
ITRRZ		GETIN	locates radial position data in CHANL array for given slice
ITSPZ		GETIN	locates pin-fin spacing data in CHANL array for given slice
ITXNZ		GETIN	locates impingement-hole spacing data in CHANL array for given slice
IUNITS	UNITS		indicates system of units used for input data
IUNSTB		FLSPLT	indicates whether flow split is stable or not
IUP		PARRAY	upstream node number for coolant-channel pressure calculations
IUPS		PARRAY	upstream station number for coolant-channel pressure calculations
IVARS(12)		PLOTMF	array of integer plotting controls
IVERGE		TCOEF WROUT	pressure iteration loop counter

Variable	Common	Subroutine	Definition
IWR		GAUSS	control on debugging output of coefficient matrix
IWRITE	SPECL		input control on amount of printed output
IXAX		PLOTMF	logical variable with value .TRUE.
IYAX		PLOTMF	locical variable with value .FALSE.
I1		PREP	starting point in INDCHN array for integer data for given slice, also used as starting point for nodal data in CHANL array for given slice
I3		PREP	starting point in CHANL array for station data
JDIS		FLOWS	number of stations that impingement flow-split station is displaced from station 1
JHCAL		INPRT	indicates type of heat transfer at given station
JLSTM		THRCON	size of wall thermal conductivity tables
JNUMS		FLSPLT	indicates whether previous call to FLSPLT resulted in unstable flow split
JOUTRG		FLSPLT	indicates attempt to split more than 100 percent of flow to one side
JPIV		GAUSS	pivotal column in matrix to be reduced
JS		FLOWS FLSPLT HCOOL PARRAY TARRAY TCOEF WROUT	station number at impingement flow-split point

Variable	Common	Subroutine	Definition
JSENS		FLOWS GETIN FLSPLT PARRAY TARRAY TCOEF	indicates which side of blade impingement flow-split is on
JSGNCH		FLSPLT	indicates whether a sign change has occurred in EPSN
JSO(15)		TCOEF	array for saving converged-flow-split station number for each slice
JSOLDS(25)		FLSPLT	array to keep track of stations checked as flow-split stations
JTIMES		FLSPLT	indicates whether DELTAN is to be rough adjusted or fine tuned
K		GAUSS	total bandwidth of matrix to be solved
K		NTTACT	counter of number of overall coolant-flow iterations
KSIG		PLNUM	coolant-plenum iteration counter
L		TARRAY THRCON	midwall node number at given station
LCOOL		FLOWS HCFRCD HCOOL HCPINS TARRAY TCOEF WROUT	coolant-channel node number at given station
LCOOLP		HCPINS TARRAY	inner-surface node across coolant-channel from given station
LCUP		HCPINS TARRAY	coolant-channel node number upstream of given station
LCUPP		TARRAY	inner-surface node across coolant channel and upstream of given station

Variable	Common	Subroutine	Definition
LCUPS		TARRAY	inner-surface node upstream of given station
LDN		TARRAY	midwall node number downstream of given station
LIN		FLOWS HCFRCD HCOOL HCPINS TARRAY	inner-surface node number at given station
LJ		INPRT TARRAY THRCON	node number at junction of outer coating and wall metal at given station
LOUT		TARRAY THRCON	wall outer-surface node number at given station
LSP		PREP	starting point in BCXP array of data for given slice
LSS		PREP	starting point in BCXS array of data for given slice
LUP		TARRAY	midwall node number upstream of given station
MACH1		PLNUM	indicator to keep track of step-size change
MD1	SPECL		indicator to control special condensed, on-line output
MD2	SPECL		indicator that job is complete and summary plots are to be produced
MD3	SPECL		plot counter
MNBC		INPRT	maximum of NBCS and NBCP
N		GAUSS	number of rows in matrix to be solved
NAG		PLNUM	indicator whether calculations have progressed beyond initial station

Variable	Common	Subroutine	Definition
NBCP	BOUND		input number of boundary condition points on pressure side
NBCS	BOUND		input number of boundary condition points on suction side
NBFRP		INPRT	number of pressure-side boundary condition points preceding data for given slice at given time
NBFRS		INPRT	number of suction-side boundary condition points preceding data for given slice at given time
NBLKSZ	TCO		size of data block in CHANL array for given slice
NCC		PLNUM	loop counter
NCHAR		NTTACT	number of characters in ALPH2 array
NCOOL		PLOTMF	coolant node number
NEND		HCOOL	end of region that uses leading-edge impingement correlation
NFC		FLOWS	number of station containing film-cooling holes
NFCSUP(80)	FLMCOL		array identifying node supplying film cooling to each downstream node
NFLUID(200)		INPRT	array of coolant-channel node numbers for each station
NFWD	TCO		number of stations in forward region
NG	GAAS		number of temperature entries in gas property table, GS
NGEO		GETIN	number of NAMELISTs /GEO/ to be read in
NINTRV		PLOTMF	number of temperature intervals in summary plots
NIT		TCOEF	counter of number of overall coolant-flow iterations

Variable	Common	Subroutine	Definition
NL		INPRT	output line counter
NLBLS		PLOTMF	number of points at which symbols are to be plotted
NMM		PLOTMF	midwall node number
NMW		NTTACT	outer-surface node number
NODM		WROUT	midwall node
NODOUT		GETIN	outer-surface node
NODSF		FLOWS GETIN INPRT PREP FLSPLT PARRAY TARRAY TCOEF	number of nodes in forward region
NODST		GETIN INPRT PREP PARRAY TARRAY TCOEF NTTACT	total number of nodes for given slice
NODSTM		INPRT	total number of nodes minus 4
NOS		FLOWS INPRT WROUT PLOTMF	outer-surface node number
NPRCP		INPRT	number of points in each pressure-side boundary condition array for times preceding current time
NPRCS		INPRT	number of points in each suction-side boundary condition array for times preceding current time

Variable	Common	Subroutine	Definition
NPRTP		INPRT	number of points in each pressure-side boundary condition array per time step
NPRTS		INPRT	number of points in each suction-side boundary condition array per time step
NPTS		PLOTMF	number of points on given plot
NROW		GAUSS	number of matrix row to be displayed by debug output
NSAVE		TCOEF	coolant node number just upstream of exit, location of TSAVE
NSLICE	TCO		current slice number
NSTA	TCO		number of stations per slice
NSTAPS		PLOTMF	number of stations on each side of blade
NSTNS		PLNUM	number of spanwise stations per slice in coolant plenum
NTBC		GETIN	number of entries in input BCTIME array
NTIMES		INPRT	number of entries in BCTIME array
NTTG		PREP NTTACT	time step number
NTYM		NTTACT	number of time steps in transient
NUMS		FLSPLT	counter to force at least four attempts at a good flow split
P(2,15,80)	TCO		pressure at each node, for two consecutive time steps, lbf/in^2
PAVG		FLOWS	coolant-channel static pressure, lbf/in^2, used in calculating impingement hole flow rates
PBAR		FLOWS	pressure used in calculating square of coolant-channel Mach number

Variable	Common	Subroutine	Definition
PCNVRG		TCOEF	pressure-difference convergence criterion
PD		FLOWS GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	Prandtl number
PDTOG		HCOOL	Prandtl number based on coolant-supply temperature
PEX(400)	BOUND		input array containing tables of static pressure at trailing-edge coolant exhaust, lbf/in^2
PEXC		INPRT	static pressure at trailing-edge coolant exhaust in SI units, kPa, for given slice
PEXIT(15)	TCO		static pressures at trailing-edge coolant exhaust for each slice at given time, lbf/in^2
PEXOLD(15)		TCOEF	saved value of exhaust static pressure, lbf/in^2, used in setting initial guess of pressure distribution for subsequent time step
PEXTT		PLNUM	total pressure at exit of coolant plenum for given slice, lbf/in^2
PG(80)	FLMCOL		array containing hot-gas-side static pressure, lbf/in^2, at each station
PI		HCOOL	constant, 3.14159
PIM	TCO		impingement-supply pressure, lbf/in^2
PIMOLD(15)		TCOEF	saved value of impingement-supply pressure, lbf/in^2, for each slice

Variable	Common	Subroutine	Definition
PIN(15)	RADL		coolant total pressure, lbf/in^2, at entrance to each slice
PINS		HCPINS TARRAY	number of pin fins at given station
PIVOT		GAUSS	main diagonal term of row of matrix being solved
PLEGN(5)		PLOTMF	alphameric array to label pressure-data plots
PLTYME(2)		PLOTMF	alphameric variable to print transient time on each plot
POLD(15,80)		TCOEF	saved values of coolant-channel pressure, lbf/in^2, from previous iteration
PP		PLNUM	intermediate pressure term
PPLEN		NTTACT	impingement-supply pressure, lbf/in^2
PROD		HCOOL	intermediate calculation result
PROPS		GETIN	NAMELIST name
PSAV(5)		TCOEF	array to save last four values of pressure at flow-split station, used to check convergence
PTEMP		PLNUM	intermediate pressure term
PTEXIT		PLNUM	coolant-plenum total pressure, lbf/in^2: entrance value going into subroutine, exit value coming out
PTIN		NTTACT	coolant-supply pressure for a givem time, lbf/in^2
PTIO(50)	BOUND		input array of coolant-supply pressure, kPa (lbf/in^2), as function of time, sec
PTIOC		INPRT	initial coolant-supply pressure in SI units, kPa
PTNOLD		NTTACT	previous value of PTIN, lbf/in^2
PT1		PLNUM	calculated coolant-plenum inlet total pressure, lbf/in^2

Variable	Common	Subroutine	Definition
PUMP(80)	TCO		term to account for coolant pumping due to wheel rotation
PUMTRM		PARRAY	term to account for coolant pumping due to wheel rotation
PXX		PLNUM	average static pressure, lbf/in^2, in coolant plenum for given slice
QG(80)	TCO		hot-gas heat flux to blade at each station, $Btu/hr \cdot ft^2$
QHUB(80)	BOUND		heat flux conducted to blade wall from hub platform at each station, $Btu/hr \cdot ft^2$
QHUBIN(400)	BOUND		input table of hub heat flux at each station as function of time, W/m^2 ($Btu/hr \cdot ft^2$)
QSNK(80)	TCO		term to account for heat removal from wall by film-cooling flow through wall
QTIP(80)	BOUND		heat flux from blade wall at tip for each station, $Btu/hr \cdot ft^2$
QTIPIN(400)	BOUND		input table of tip heat flux at each station as function of time, W/m^2 ($Btu/hr \cdot ft^2$)
R	TCO		gas constant; value for air is build in, 53.35 ft-lbf/lbm $\cdot$ ^{o}R
RATIO		THRCON	interpolating fraction
RCHRD		TARRAY	dimensionless ratio of time increment to chordwise length increment squared at each station
RCHRDM		TARRAY	maximum value of RCHRD for a given slice
RCVRY		TARRAY	recovery factor
RE(80)	PRPS		coolant-channel Reynolds number at each station

Variable	Common	Subroutine	Definition
REFC(80)	FLMCOL		film-cooling flow Reynolds number at each station
REJ(80)		HCOOL	impingement-jet Reynolds number at each station
REJOVR(80)		HCOOL	array to save values of impingement-jet Reynolds number that are out of range of correlation
REY		PLNUM	coolant-plenum Reynolds number based on hydraulic diameter
RHOBAR		TARRAY	mean density in coolant channel, lbm/in^3
RHOC	TRNSNT		input density of outer coating, kg/m^3 (lbm/ft^3)
RHOM	TRNSNT		input density of wall metal, kg/m^3 (lbm/ft^3)
RHOVG(400)	BOUND		input table of hot-gas-side, free-stream mass velocity at each station as function of time, $kg/m^2 \cdot sec$ ($lbm/ft^2 \cdot sec$)
RHOVGA(80)	FLMCOL		hot-gas-side, free-stream mass velocity at each station for given slice, $lbm/ft^2 \cdot sec$
RI		GETIN	input value of radial location of coolant-plenum inlet for given slice, cm (in.)
RIN(15)	RADL		table of RI values for each slice, in.
RO		GETIN	input value of radial location of coolant-plenum exit for given slice, cm (in.)
ROINVC		HCOOL	intermediate term in impingement correlation, ft^3/lbm
ROINVJ		HCOOL	intermediate term in impingement correlation, ft^3/lbm
ROOT		PARRAY	intermediate term in pressure calculations

Variable	Common	Subroutine	Definition
ROUT(15)	RADL		table of RO values for each slice, in.
RR(80)	TCO		mean radial location of each station for a given slice
RRP		PLNUM	radial location, in.
RTEMP		PLNUM	radial location, in.
RTNARR(2)		PLOTMF	array containing maximum and minimum values of plot variables
RTRNV		TARRAY	dimensionless ratio of time increment to through-the-wall length increment squared
RTRNVM		TARRAY	maximum value of RTRNV for given slice
S(15)	TCO		span of each slice, in.
SEGMTS		PLNUM	number of segments in coolant plenum for given slice
SIGB		PLNUM	dummy variable used in coolant-plenum calculations
SIGC		PLNUM	dummy variable used in coolant-plenum calculations
SIGMA(20)		PLNUM	coolant velocity distribution in coolant plenum
SLEGN(5)		PLOTMF	alphameric array to label suction-side plots
SLP		HCPINS	mean pin-fin length at given station, in.
SP(80)	PRPS		pin-fin spacing at each station, in.
SPAN	TCO		radial span of given slice, in.
SPANC		INPRT	radial span of given slice in SI units, cm
SPHTC	TRNSNT		input specific heat of outer coating, $J/kg \cdot K$ ($Btu/lbm \cdot {}^{o}F$)
SPHTM	TRNSNT		input specific heat of wall metal, $J/kg \cdot K$ ($Btu/lbm \cdot {}^{o}F$)

Variable	Common	Subroutine	Definition
ST		HCOOL	Stanton number calculated from user-supplied impingement correlation
STANMX		HCOOL	Stanton number calculated from leading-edge impingement correlation
SV(3)		PLNUM	array to save values of SIGC
SYMBL(10)		PLOTMF	array of integers to be used as plot symbols
SYMBOL		PLOTMF	particular entry from SYMBL array
T(2, 15, 400)	TCO		calculated temperature at each node for each slice for two time steps, ^{o}F
TABOVE		TARRAY	midwall temperature at given station in slice above current slice, ^{o}F
TAU(400)	TCO		array of thickness values, in.
TBAR		FLOWS	coolant temperature, ^{o}R, used to calculate Mach number
TBAR		HCPINS	ratio of temperature drops in pin fins, pressure-side wall temperature minus mid-coolant-channel temperature to suction-side wall temperature minus mid-coolant-channel temperature
TBAR		WROUT	mean outer-surface temperature for given slice, ^{o}F
TBARMD		WROUT	mean midwall temperature, ^{o}F
TBELOW		TARRAY	midwall temperature at given station in slice below current slice, ^{o}F
TBULK		WROUT	overall blade bulk-metal temperature, ^{o}F
TC		THRCON	mean temperature of blade cladding material, ^{o}F
TCIN		NTTACT	coolant temperature, ^{o}F
TCOF(400, 30)	MATRX		array of coefficients to be solved for temperature or pressure

Variable	Common	Subroutine	Definition
TDHF		GETIN	function of film-cooling-hole size and spacing
TDHJ		GETIN	hydraulic diameter of impingement hole
TDLX(5)		GETIN	array containing lengths from nodes at upstream station to corresponding nodes at current station
TDP		GETIN	input pin-fin diameter, cm (in.)
TEM		INPRT	coolant-inlet absolute temperature, K ($^\circ$R)
TEPS	TRNSNT		factor used to define time mean properties
TERM		FLSPLT	adjustment to flow-split parameter
TG(80)	TCO		hot-gas reference temperature at each station for given slice, $^\circ$R
THETA1		TARRAY	intermediate terms in nodal energy equations
THETA2			
THETA3			
THETA4			
THETA5			
THETA6			
THETA8			
THETA9			
THK(3)		GETIN	input array of thickness values, cm (in.)
THUB(80)	BOUND		specified temperature at each blade hub station for given time, $^\circ$F
THUBIN(400)	BOUND		input table of hub temperatures at each station as function of time, K ($^\circ$F)
TIKLE(30)		GETIN	blank alphameric array used to initialize title array

Variable	Common	Subroutine	Definition
TIN(15)	RADL		inlet total temperature in coolant plenum for each slice, ^{o}F
TIP1		TARRAY	term in conduction equation to account for specified tip temperature
TIP3		TARRAY	term in conduction equation to account for specified tip heat flux
TITL		GETIN	NAMELIST name
TITLE(30)	SPECL		alphameric array of title information from input
TLABL1(21) TLABL2(9)		PLOTMF	alphameric arrays used to put input title on plots
TMAXP		PLOTMF	maximum pressure-side temperature to be plotted
TMAXS		PLOTMF	maximum suction-side temperature to be plotted
TMFRAC		PREP	time-interpolating function
TMINP		PLOTMF	minimum pressure-side temperature to be plotted
TMINS		PLOTMF	minimum suction-side temperature to be plotted
TMP		GASTBL	temperature used to determine gas properties, ^{o}R
TMP1		GASTBL	temperature used to determine gas properties, ^{o}F
TOTSPN		WROUT	total span of blade, in.
TP		PLNUM	intermediate temperature variable
TPLEN		NTTACT	mean impingement-plenum static temperature, ^{o}F
TPM(500)		PLOTMF	table of pressure-side, midwall temperatures for summary plots, K (^{o}F)
TPMAX		WROUT	maximum pressure-side, outer-surface temperature for given slice, K (^{o}F)

Variable	Common	Subroutine	Definition
TPMIN		WROUT	minimum pressure-side, outer-surface temperature for given slice, K ($^{\circ}$F)
TPO(500)		PLOTMF	table of pressure-side, outer-surface temperatures for summary plots, K ($^{\circ}$F)
TREDGE		TARRAY	intermediate term in nodal energy equation in trailing-edge region
TREPS		PARRAY TARRAY	same as TEPS
TRR		GETIN	input mean radial location of given station, cm (in.)
TRTRM		PARRAY	intermediate transient term in pressure equations
TRTRMC		TARRAY	intermediate transient term involving outer coating
TRTRMG		TARRAY	intermediate transient term involving coolant
TRTRMJ		TARRAY	intermediate transient term for coolant channel at entrance to trailing edge
TSAVE		TCOEF	temperature in coolant channel, just upstream of exit
TSM(500)		PLOTMF	table of suction-side, midwall temperature for summary plots, K ($^{\circ}$F)
TSMAX		WROUT	maximum suction-side, outer-surface temperature for given slice, K ($^{\circ}$F)
TSMIN		WROUT	minimum suction-side, outer-surface temperature for given slice, K ($^{\circ}$F)
TSO(500)		PLOTMF	table of suction-side, outer-surface temperatures for summary plots, K ($^{\circ}$F)
TSP		GETIN	input pin-fin spacing, cm (in.)
TTEMP		PLNUM	variable to save temperature value, $^{\circ}$R

Variable	Common	Subroutine	Definition
TTEXIT		PLNUM	coolant total temperature in plenum, ^{O}F
TTIN		NTTACT	coolant-supply temperature, ^{O}F, at entrance to coolant plenum for given time
TTIO(50)	BOUND		input table of coolant-supply temperature, K (^{O}F), as function of time
TTIP(80)	BOUND		table of blade tip temperature, ^{O}F, for given time
TTIPIN(400)	BOUND		input table of blade tip temperature, K (^{O}F), as function of time
TTOTC(80)		TCOEF	coolant total temperature at each station for given slice, ^{O}F
TTX		PLNUM	coolant total temperature at inlet to coolant plenum for given slice, ^{O}R
TTYME		WROUT	current time in transient, sec
TT1(20)		PLNUM	total-temperature distribution, ^{O}F, in coolant plenum
TW		THRCON	midwall temperature for evaluating thermal conductivity
TXN		GETIN	input spanwise spacing of impingement jets at given station, cm (in.)
TXX		PLNUM	average static temperature in coolant plenum for given slice, ^{O}F
TYME	TRNSNT		time in transient calculations, sec
TYMMAX	TRNSNT		maximum time to which transient is carried, sec
T0G	TCO		impingement-jet temperature for given slice, ^{O}R
T1		PLNUM	inlet total temperature in coolant plenum for given slice, ^{O}R
UA(2)		INPRT	alphameric array containing area units

Variable	Common	Subroutine	Definition
UL(2)		INPRT	alphameric array containing length units
V		TCOEF	factor to accelerate convergence of pressure iterations
VARIB(15)		PLOTMF	alphameric array containing some plot labels
VARS(12)		PLOTMF	array containing plotting controls
VDP		HCPINS	pin-fin diameter at given station, in.
VOLBAR		TARRAY	coolant-channel volume element at entrance to trailing-edge region, in^3
VSP		HCPINS	pin-fin spacing at given station, in.
V1		PLNUM	intermediate term in coolant-plenum calculations
W(15)	RADL		coolant flow-rate at entrance to each slice, lbm/hr
WC		HCOOL	absolute value of coolant flow-rate, lbm/sec
WCHK(80)	CHKHOL		alphameric variable used to indicate choked flow in impingement jets
WCHKDM	CHKHOL		alphameric variable used to indicate choked flow in holes dumping coolant to trailing-edge region
WCHOKE		PLNUM	coolant flow-rate at entrance to given slice, lbm/hr
WCR		FLOWS	critical flow-rate, lbm/sec
WCROS(2, 15, 80)	TCO		coolant-channel crossflow rate at each station for each slice for two time steps, lbm/sec
WDUMP	TCO		rate of coolant flow being dumped directly from plenum to trailing-edge region, lbm/sec
WFC(80)	TCO		film-cooling flow rate at each station for given slice, lbm/sec

Variable	Common	Subroutine	Definition
WFCDUM		FLOWS	intermediate variable in film-cooling flow rate calculation
WFCDUM		PARRAY	total film-cooling flow rate at given station, lbm/sec
WIM	TCO		total impingement-jet flow rate for given slice, lbm/sec
WJ(15,80)	TCO		impingement-jet flow rate for each station for each slice, lbm/sec
WPLEN	BOUND		input initial guess at total coolant flow, kg/hr (lbm/hr)
WPLENC		INPRT	total coolant flow in SI units, kg/hr
WPLENO		NTTACT	variable to save previous estimate of total coolant flow, lbm/hr
WS	RADL		rotor speed at given time, rpm
WSVST(50)	BOUND		input table of rotor speed, rpm, versus time, sec
WUSED		NTTACT	cumulative amount of coolant used, up to current slice, lbm/hr
WXCP		TARRAY	coolant flow-rate times specific-heat term
WXX		PLNUM	coolant flow-rate at entrance to given slice, lbm/hr
X(80)		TCOEF	coolant-channel node locations, in., used to set initial pressure distribution
XBAR		FLOWS	term in film-cooling effectiveness correlation
XCC		INPRT	coolant-channel distance from station 1 to given station, cm (in.)
XDUM		FLOWS	dummy variable used to save location of film-cooling hole

Variable	Common	Subroutine	Definition
XFC(80)	FLMCOL		distance from station with film-cooling holes to downstream stations without film-cooling holes, in.
XIS		INPRT	inside-wall surface distance from station 1 to given station
XJN		INPRT	distance along junction of cladding and wall metal from station 1 to given station, cm (in.)
XK(4)		PLNUM	pressure change in coolant plenum
XL		HCOOL	length of inner surface used in leading-edge impingement correlation, in.
XL(4)		PLNUM	temperature change in coolant plenum
XLABL(29)		PLOTMF	alphameric array of plot labels
XLABL2(15)		PLOTMF	alphameric array of plot labels
XLBL(20)		PLOTMF	array of x-coordinates to be plotted as slice numbers
XMM		INPRT	midwall distance from station 1 to given station, cm (in.)
XMU		FLOWS GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	viscosity, lbm/ft · hr
XMUC(80)	FLMCOL		coolant viscosity, lbm/ft · hr, at each station, evaluated at mean temperature between inner-wall surface and bulk coolant temperatures
XMUM		FLOWS	hot-gas viscosity, lbm/ft · hr
XMUTOG		HCOOL	coolant viscosity based on coolant-supply temperature, lbm/ft · hr

Variable	Common	Subroutine	Definition
XN(80)	TCO		spanwise spacing of impingement holes at each station, in.
XNN		PLNUM	factor for increasing number of stations in coolant plenum
XOD		PREP	ratio of impingement-hole spacing to hydraulic diameter, at given station
XOS		INPRT	wall outer-surface distance from station 1 to given station, cm (in.)
XOVERD		HCOOL	ratio of impingement-hole spacing to hydraulic diameter, at given station
XOVRL		HCPINS	location of zero temperature gradient in pin fins
XP		HCOOL	length of pressure-side inner-wall surface in leading-edge impingement region
XP(80)		PLOTMF	pressure-side, dimensionless distance along midwall plane from station 1 to each station
XP		PREP	distance of given pressure-side station from station 1, in.
XPF		PREP	interpolating fraction in BCXP table
XPL		PLOTMF	overall length along pressure-side, midwall plane, cm (in.)
XS		HCOOL	length of suction-side, inner-wall surface in leading-edge impingement region
XS(80)		PLOTMF	suction-side, dimensionless distance along midwall plane from station 1 to each station
XS		PREP	distance of given suction-side station from station 1, in.
XSF		PREP	interpolating function in BCXS table

Variable	Common	Subroutine	Definition
XSL		PLOTMF	overall length along suction-side, mid-wall plane, cm (in.)
XTEST		PLNUM	convergence test variable
XTOT		WROUT	overall outer-surface length around blade, in.
XTOTMD		WROUT	overall midwall length around blade, in.
XXN		PLNUM	number of spanwise stations per slice in coolant plenum
Y(320)		PLOTMF	array containing temperature values to be plotted
YCNVUU		TARRAY	indicates forced-convection heat transfer at last forward-region station on pressure side
YCONV		TARRAY	indicates forced-convection heat transfer at given station
YCONVU		TARRAY	indicates forced-convection heat transfer at station immediately upstream of given station
YFINS		TARRAY	indicates pin-fin heat transfer at given station
YFINSU		TARRAY	indicates pin-fin heat transfer at station immediately upstream of given station
YFNSUU		TARRAY	indicates pin-fin heat transfer at last forward-region station on pressure side
YIMP		TARRAY	indicates impingement heat transfer at given station
YIMPU		TARRAY	indicates impingement heat transfer at station immediately upstream of given station
YIMPUU		TARRAY	indicates impingement heat transfer at last forward-region station on pressure side

Variable	Common	Subroutine	Definition
YLABL(7)		PLOTMF	alphameric array for labeling plots
YLABL2(11)		PLOTMF	alphameric array for labeling plots
YLBL(20)		PLOTMF	array of coordinates of points to be plotted as slice numbers
YMAX		PLOTMF	maximum value of y-coordinates on plot
YMIN		PLOTMF	minimum value of y-coordinates on plot
YPLABL(10)		PLOTMF	alphameric array for labeling plots
YTEM(80)		PLOTMF	array to be plotted
ZED		PLNUM	coolant-plenum pressure-drop parameter for given slice
ZOVERD		HCOOL	ratio of coolant-channel width to impingement-hole hydraulic diameter
Z1(15)		PLNUM	coolant-plenum pressure-drop parameter for each slice
Z3		PLNUM	intermediate term involving coolant flow
Z4		PLNUM	intermediate term involving coolant flow

```
C----SOURCE.NTTACT---THIS IS THE MAIN PROGRAM.  BLOCK DATA SUBPROGRAM
C                    GASDAT MUST BE LOADED FIRST.
C
C           TRANSIENT THERMAL ANALYSIS OF A COOLED TURBINE BLADE
C           *                *             *      *
C                                       TACT1
C
      COMMON /BOUND/ BCXS(400),    BCXP(400),    BCHGS(1000), BCHGP(1000),
     Z               BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),
     Z               BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),
     Z               QHUBIN(400), QHUB(80),    TTIPIN(400), TTIP(80),
     Z               QTIPIN(400), QTIP(80),    RHOVG(400),  PEX(400),
     Z               BCTIME(50),  TTIO(50),    PTIO(50),    WPLEN,
     Z               WSVST(50),   AKCTBL(20),  AKWTBL(20),  NBCS,   NBCP
C
      COMMON /FLMCOL/ RHOVGA(80), PG(80),    XFC(80),   FLMEFF(80),
     Z                XMUC(80),   EMES(80),  REFC(80),  NFCSUP(80)
C
      COMMON /GAAS/ GS(200),   NG
C
      COMMON /RADL/ APLN(15),  DPLN(15),   RIN(15),    ROUT(15),
     Z              PIN(15),   TIN(15),    W(15),      WS
C
      COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),
     Z                IPLOT,  MD1,  MD2,  MD3, IADJIN, IWRITE
C
      COMMON /TCO/ ADUMP,     BTA,        CD,          CP,
     Z             GAM,       PIM,        R,           SPAN,        TOG,
     Z             WDUMP,     WIM,        AKC(15,80),  AKW(15,80),
     Z             A(400),    AJET(80),   AM2(80),     CNUM(80),
     Z             DH(80),    DHF(80),    DHJ(80),
     Z             DLX(400),  FF(80),     HC(80),      HG(80),
     Z             P(2,15,80),PEXIT(15),  PUMP(80),    QG(80),
     Z             QSNK(80),  RR(80),     S(15),       T(2,15,400),
     Z             TG(80),    TAU(400),   WFC(80),
     Z             WJ(15,80), WCROS(2,15,80),          XN(80),
     Z             ICOR,      IFILM,      IHUB,        ITIP,
     Z             ISBLOK,    ISLICE,     NBLKSZ,      NSLICE,
     Z             NFWD,      NSTA,       IHC(80)
C
      COMMON /TRNSNT/ RHOC,      RHOM,      SPHTC,      SPHTM,
     Z                DLTYME,    TYME,      TEPS,       TYMMAX
C
      COMMON /UNITS/ CINCH(2), CHTC(2),  CHFLX(2), CPRSR(2), CMSFL(2),
     Z               CTMPF(2), CTCON(2), CDEN(2),  CSPHT(2), CGASC(2),
     Z               CVISC(2), CRHOVG(2),IUNITS
C
      DIMENSION DP(80), SP(80), ALPH(12), ALPH2(4), CD1(200)
C
C    TTIO = TOTAL TEMPERATURE OF BLADE COOLING AIR AT INLET
C    WPLEN = ESTIMATE OF COOLANT FLOW RATE - USED AS FIRST GUESS
C    PTIO = TOTAL PRESSURE OF BLADE COOLING AIR AT INLET
C    PEX = EXTERNAL GAS STREAM STATIC PRESSURE AT TRAILING EDGE
C
      DATA ALPH/' THI','S JO','B WA','S ST','ARTE','D AT',
     Z          '    ','    ',',ON ','    ','    ','    '/
      DATA NCHAR/16/
C
      MD1 = 0
      MD2 = 0
```

```
      TYME = -1.0                                                        NTTACT--0061
      DLTYME = 0.0                                                       NTTACT--0062
C                                                                        NTTACT--0063
C  TO GET AN ABBREVIATED OUTPUT OF MID-WALL TEMPERATURES AT THE TERMINALNTTACT--0064
C    FOR EACH SLICE, ENTER:                                              NTTACT--0065
C AT TACT.50;SET TACT.MD1=1                                              NTTACT--0066
C                                                                        NTTACT--0067
      MD3 = 0                                                            NTTACT--0068
      K = 1                                                              NTTACT--0069
C                                                                        NTTACT--0070
C   RECORD STARTING TIME, TO BE USED TO IDENTIFY MICROFILM PLOTS         NTTACT--0071
C                                                                        NTTACT--0072
      CALL TIME(NCHAR,ALPH2)                                             NTTACT--0073
      ALPH(7) = ALPH2(3)                                                 NTTACT--0074
      ALPH(8) = ALPH2(4)                                                 NTTACT--0075
      ALPH(10) = ALPH2(1)                                                NTTACT--0076
      ALPH(11) = ALPH2(2)                                                NTTACT--0077
      WRITE(6,425) (ALPH(I),I=1,12)                                      NTTACT--0078
      WRITE(8,425) (ALPH(I),I=1,12)                                      NTTACT--0079
C                                                                        NTTACT--0080
C   READ IN DATA                                                         NTTACT--0081
C                                                                        NTTACT--0082
      CALL GETIN(IWRITE,TYMMAX,WSVST,IADJIN)                             NTTACT--0083
C                                                                        NTTACT--0084
C   WRITE TITLE PAGE                                                     NTTACT--0085
C                                                                        NTTACT--0086
      WRITE(6,400)                                                       NTTACT--0087
      WRITE(6,425) (ALPH(I),I=1,12)                                      NTTACT--0088
      WRITE(6,430) (TITLE(I),I=1,30)                                     NTTACT--0089
400   FORMAT(1H1,//////,50X,'***** OUTPUT *****',/////)                  NTTACT--0090
425   FORMAT(/36X,12A4)                                                  NTTACT--0091
430   FORMAT(//1X,30A4)                                                  NTTACT--0092
C                                                                        NTTACT--0093
      TTIN = TTIO(1)                                                     NTTACT--0094
      PTIN = PTIO(1)                                                     NTTACT--0095
440   WPLENO = WPLEN                                                     NTTACT--0096
      PTNOLD = PTIN                                                      NTTACT--0097
      PTIO(1) = PTIN                                                     NTTACT--0098
      TYME = 0.0                                                         NTTACT--0099
      NTYM = 1                                                           NTTACT--0100
      IF (DLTYME.GT.0.) NTYM = TYMMAX/DLTYME + 1                         NTTACT--0101
      NODST = 5*NSTA                                                     NTTACT--0102
C                                                                        NTTACT--0103
C                                                                        NTTACT--0104
C  START MARCHING                                                        NTTACT--0105
C                                                                        NTTACT--0106
      DO 1100 ITYM = 1,NTYM                                              NTTACT--0107
      ITYME = ITYM-1                                                     NTTACT--0108
      NTTG = ITYM                                                        NTTACT--0109
      TYME = ITYME*DLTYME                                                NTTACT--0110
      IF (ITYM.EQ.1) TYME = -1.                                          NTTACT--0111
C                                                                        NTTACT--0112
C                                                                        NTTACT--0113
C                                                                        NTTACT--0114
C-- EVALUATE TIME DEPENDENT BOUNDARY CONDITIONS -----------------------NTTACT--0115
C     -                                                                  NTTACT--0116
C                                                                        NTTACT--0117
      PTIN = PTIO(1)                                                     NTTACT--0118
      IF (TYME.LT.0.0) GO TO 490                                         NTTACT--0119
C                                                                        NTTACT--0120
```

```
C-- LOCATE COOLANT SUPPLY PRESSURE FOR TYME
C
      DO 450 I = 4,50,2
      PTIN = PTIO(I-3)
      IF (PTIO(I).LE.0.0)  GO TO 460
      IPTIO = I-1
      IF (TYME.LE.PTIO(I).AND.TYME.GT.PTIO(I-2))  GO TO 455
450   CONTINUE
455   DENOM = PTIO(IPTIO+1)-PTIO(IPTIO-1)
      IF (DENOM.GT.0.)  PTIN = PTIO(IPTIO-2) +
     Z              (PTIO(IPTIO)-PTIO(IPTIO-2))*(TYME-PTIO(IPTIO-1))/DENOM
460   CONTINUE
C
C-- LOCATE COOLANT SUPPLY TEMPERATURE FOR TYME
C
      DO 470 I = 4,50,2
      TTIN = TTIO(I-3)
      IF (TTIO(I).LE.0.0)  GO TO 490
      ITTIO = I-1
      IF (TYME.LE.TTIO(I).AND.TYME.GT.TTIO(I-2))  GO TO 475
470   CONTINUE
475   DENOM = TTIO(ITTIO+1)-TTIO(ITTIO-1)
      IF (DENOM.GT.0.)  TTIN = TTIO(ITTIO-2) +
     Z              (TTIO(ITTIO)-TTIO(ITTIO-2))*(TYME-TTIO(ITTIO-1))/DENOM
490   CONTINUE
C
C  LOCATE THE VALUE OF THE WHEEL SPEED FOR THE CURRENT TIME.
C
      WS = WSVST(1)
      IF (TYME.LE.0.0)  GO TO 530
      DO 510 I = 4,50,2
      WS = WSVST(I-3)
      IF (WSVST(I).LE.0.0)  GO TO 530
      IWS = I-1
      IF (TYME.LE.WSVST(I).AND.TYME.GT.WSVST(I-2))  GO TO 520
510   CONTINUE
520   DENOM = WSVST(IWS+1)  -  WSVST(IWS-1)
      IF (DENOM.GT.0.0)  WS  =  WSVST(IWS-2)  +
     Z              (WSVST(IWS)-WSVST(IWS-2))*(TYME-WSVST(IWS-1))/DENOM
530   CONTINUE
C
C
C  LOCATE THE VALUE FOR PEXIT, GAS STATIC PRESSURE AT EXIT OF BLADE,
C   FOR THE CURRENT TIME AND ALL SLICES.
C
      IF (TYME.GT.0.0)  GO TO 533
      DO 532 I = 1,NSLICE
      PEXIT(I)  = PEX(I)
      IF (PEX(I).LE.0.0)  PEXIT(I)  = PEX(1)
532   CONTINUE
533   CONTINUE
      IF (BCTIME(2).LE.0.0)  GO TO 545
      DO 535 I = 2,50
      IPEX = I-1
      IF (TYME.LE.BCTIME(I).AND.TYME.GT.BCTIME(I-1))  GO TO 540
535   CONTINUE
540   DENOM = BCTIME(IPEX+1)  -  BCTIME(IPEX)
      IF (DENOM.EQ.0.0)  GO TO 545
      TYMFRC = (TYME - BCTIME(IPEX))/DENOM
      DO 542 I = 1,NSLICE
```

```
      IC = (IPEX-1)*NSLICE + I
      PEXIT(I) = PEX(IC) + (PEX(IC+NSLICE)-PEX(IC))*TYMFRC
 542  CONTINUE
      IF (IFILM.NE.2) GO TO 545
C
C-- SET INTERPOLATED VALUES OF FREE STREAM RHO*V FOR THIS TIME
C
      DO 543 I = 1,NSTA
      IRO = (IPEX-1)*NSTA + I
      IRN = IPEX*NSTA + I
 543  RHOVGA(I) = RHOVG(IRO) + TYMFRC*(RHOVG(IRN)-RHOVG(IRO))
C
C
 545  CONTINUE
C
C
C-- SET TIME INTERPOLATED VALUES OF QHUB OR THUB AND QTIP OR TTIP.
      IF (BCTIME(2).LE.0.0) GO TO 555
      DO 550 I = 1,NSTA
      IQO = (IPEX-1)*NSTA + I
      IQN = IPEX*NSTA + I
      IF (IHUB.EQ.1) THUB(I) = THUBIN(IQO) +
     Z                         TYMFRC*(THUBIN(IQN)-THUBIN(IQO))
      IF (IHUB.EQ.3) QHUB(I) = QHUBIN(IQO) +
     Z                         TYMFRC*(QHUBIN(IQN)-QHUBIN(IQO))
      IF (ITIP.EQ.1) TTIP(I) = TTIPIN(IQO) +
     Z                         TYMFRC*(TTIPIN(IQN)-TTIPIN(IQO))
      IF (ITIP.EQ.3) QTIP(I) = QTIPIN(IQO) +
     Z                         TYMFRC*(QTIPIN(IQN)-QTIPIN(IQO))
 550  CONTINUE
      GO TO 565
C
  555 CONTINUE
      DO 560 I = 1,NSTA
      IF (IHUB.EQ.1) THUB(I) = THUBIN(I)
      IF (IHUB.EQ.3) QHUB(I) = QHUBIN(I)
      IF (ITIP.EQ.1) TTIP(I) = TTIPIN(I)
      IF (ITIP.EQ.3) QTIP(I) = QTIPIN(I)
  560 CONTINUE
C
  565 TCIN = TTIN
      IF (ITYM.GT.1) WPLEN = WUSED*PTIN/PTNOLD
 570  WPLENO = WPLEN
      PTNOLD = PTIN
      WUSED = 0.0
C
C   CALCULATE TEMPERATURE AND PRESSURES FOR EACH SLICE OF THE BLADE
C
      DO 1000 I = 1,NSLICE
      ISLICE = I
C
C  FIRST DETERMINE IMPINGEMENT PLENUM CONDITIONS
C
      CALL PLNUM(WPLEN,PPLEN,PTIN,TPLEN,TCIN)
      TOG = TPLEN + 460.
      PIM = PPLEN
      IF (IUNITS.EQ.1) GO TO 860
      WRITE(6,800) I,PIM,TOG
 800  FORMAT(1H2//10X,100('*')//30X,'THE IMPINGEMENT PLENUM CONDITIONS'
     Z        ' FOR SLICE NO.',I2,' ARE:  '//60X,'PIM = ',F7.2,
```

```
     Z          ' PSIA,'//60X,'TOG = ',F7.2,' DEG. R'//10X,100('*'))
C
      IF (I.EQ.1) WRITE(6,850) WPLEN
850   FORMAT(1H ,//30X,'CENTRAL PLENUM FLOW IS ',F6.1,' LBM/HR',//)
      GO TO 890
C
860   CTOG = TOG/1.8
      CPIM = PIM/CPRSR(1)
      WRITE(6,870) I,CPIM,CTOG
870   FORMAT(1H2//10X,100('*')//30X,'THE IMPINGEMENT PLENUM CONDITIONS'
     Z          ' FOR SLICE NO.',I2,' ARE:  '//60X,'PIM = ',F7.2,
     Z          ' KPA, '//60X,'TOG = ',F7.2,' K      '//10X,100('*'))
      CWPLEN = WPLEN/CMSFL(1)
      IF (I.EQ.1) WRITE(6,880) CWPLEN
880   FORMAT(1H ,//30X,'CENTRAL PLENUM FLOW IS ',F6.1,' KG/HR',//)
C
890   CONTINUE
C
C   THEN COMPUTE TEMPERATURES AND PRESSURES
C
      CALL PREP(I,NTTG)
C
      CALL TCOEF(IWRITE,WS,K,IPLOT,ALPH2)
      IF (IPLOT.GT.0) CALL PLOTMF(ALPH2)
      IF (MD1.EQ.0) GO TO 975
C
C THIS BLOCK PRINTS SPECIAL CONDENSED OUTPUT TO THE TERMINAL IF MD1 > 0
C
      IC1 = 0
      DO 955 II = 1,NSTA,2
      IC1 = IC1 + 1
      NMW = 5*II - 4
955   CD1(IC1) = T(2,I,NMW) - 460.
      DO 960 II = 2,NSTA,2
      IC1 = IC1 + 1
      NMW = 5*II-4
960   CD1(IC1) = T(2,I,NMW) - 460.
      WRITE(8,962) I,K
962   FORMAT(/'  BLADE SLICE ',I2,',  OVERALL FLOW LOOP ',I2,
     Z         ',  SURFACE TEMPERATURE, (F), STARTING AT LEADING EDGE'/)
      INUM = NSTA/2 + 1
      WRITE(8,964) (CD1(II),II=1,INUM)
964   FORMAT(' PRESSURE SIDE'/10(2X,F7.1))
      INUM = INUM + 1
      WRITE(8,966) CD1(1),(CD1(II),II=INUM,NSTA)
966   FORMAT(//'  SUCTION SIDE'/10(2X,F7.1))
C
C
975   CONTINUE
C
C   CHECK HOW MUCH PLENUM FLOW IS LEFT
C
      WUSED = WUSED + 3600.*WIM
      EXCESW = WPLEN - 3600.*WIM
      IF (EXCESW.GT.0..AND.I.LT.NSLICE) WPLEN = EXCESW
      IF (EXCESW.LT.0..AND.I.LT.NSLICE) WRITE(6,985) I
985   FORMAT(/'  RAN OUT OF MASS FLOW IN BRANCH NO. ',I2)
1000  CONTINUE
C
C
```

```
      IF (IUNITS.EQ.2) WRITE(6,1010) WUSED
 1010 FORMAT(/30X,'AMOUNT OF COOLANT ACTUALLY USED AT THIS TIME'
     Z          ' STEP IS ',F6.1,' LBM/HR')
      CWUSED = WUSED/CMSFL(1)
      IF (IUNITS.EQ.1) WRITE(6,1011) CWUSED
 1011 FORMAT(/30X,'AMOUNT OF COOLANT ACTUALLY USED AT THIS TIME'
     Z          ' STEP IS ',F6.1,' KG/HR')
C
      DO 1040 I = 1,NSLICE
      DO 1020 J = 1,NODST
 1020 T(1,I,J) = T(2,I,J)
      DO 1040 J = 1,NSTA
 1040 P(1,ISLICE,J) = P(2,ISLICE,J)
      IF (TYME.GT.0.0) GO TO 1100
C
C--->  ADJUST COOLANT FLOW AND RECALCULATE TEMPERATURES, ETC. FOR STEADY
C      STATE CASE OR TIME =0.0
C
      EXCESW = WPLENO - WUSED
      IF (ABS(EXCESW).LT..01*WPLENO) GO TO 1100
      IF (IADJIN.GT.0) GO TO 1050
C
C ---> NORMAL ADJUSTMENT IS ON WPLEN
C
      WPLEN = WPLENO - .995*EXCESW
      K = K + 1
      PTIN = PTNOLD
      TCIN = TTIN
      GO TO 570
C
C ----> SPECIAL CASE, FOR IADJPT > 0, ADJUSTMENT IS ON PTIN
C
 1050 PTIN = PEXIT(1) + (PTNOLD-PEXIT(1))*(WPLENO/WUSED)**1.6
      WPLEN = WPLENO
      TCIN = TTIN
      K = K + 1
      GO TO 570
C
 1100 CONTINUE
C
C
      IF (IUNITS.EQ.1) GO TO 3860
 3000 WRITE(6,3500) K,WPLENO
 3500 FORMAT(/,20X,80('-'),/,23X,I2,' LOOP(S) ON INITIAL COOLANT FLOW'
     Z          ' WERE USED. FINAL VALUE IS ',F7.2,' LB/HR')
      WRITE(6,3600) EXCESW
 3600 FORMAT(5X,'RESIDUAL COOLING AIR FLOW IS ',F9.4,' LBM/HR',/
     Z          20X,80('-'))
      WRITE(6,425) (ALPH(I),I=1,12)
C
      GO TO 3900
C
 3860 CWPLEN = WPLENO/CMSFL(1)
      CEXCSW = EXCESW/CMSFL(1)
      WRITE(6,3870) K,CWPLEN
 3870 FORMAT(/,20X,80('-'),/,23X,I2,' LOOP(S) ON INITIAL COOLANT FLOW'
     Z          ' WERE USED. FINAL VALUE IS ',F7.2,' KG/HR')
      WRITE(6,3880) CEXCSW
 3880 FORMAT(5X,'RESIDUAL COOLING AIR FLOW IS ',F9.4,' KG/HR',/
     Z          20X,80('-'))
```

```
      WRITE(6,425) (ALPH(I),I=1,12)
C
3900  CONTINUE
      MD2 = 1
      IF (NSLICE.GT.1.AND.IPLOT.GT.0) CALL PLOTMF(ALPH2)
      WRITE(6,425) (ALPH(I),I=1,12)
      STOP
      END


C----SOURCE.NFLOEST
      SUBROUTINE FLOWS(JS,DELTAN,ICHOKE,AMCHOK)
C
C=========================================================================
C
C-    SOURCE.NFLOEST--
C
C  THIS SUBROUTINE COMPUTES THE FOLLOWING---
C   WJ, IMPINGEMENT JET FLOW RATES (LBM/SEC).
C   WCROS, THE CHANNEL CROSSFLOWS (LBM/SEC).
C   AM2,  THE SQUARE OF THE CHANNEL MACH NUMBER.
C   WDUMP, A FLOWRATE DUMPED DIRECTLY FROM CENTRAL PLENUM INTO TRAILING
C          EDGE CHANNEL (LBM/SEC).
C   WIM,  THE TOTAL COOLANT FLOW USED FOR THIS SLICE (LBM/SEC).
C   WFC,  FILM COOLING HOLE FLOW RATES  (LBM/SEC).
C   FF,  THE CHANNEL FRICTION FACTOR, EITHER FOR LAMINAR, TURBULENT, OR
C        A PIN FIN ARRAY.
C   FLMEFF,  A FILM COOLING EFFECTIVENESS.
C
C=========================================================================
C
      COMMON /CHKHOL/  WCHK(80),  WCHKDM
C
      COMMON /FLMCOL/ RHOVGA(80), PG(80),    XFC(80),  FLMEFF(80),
     Z                XMUC(80),   EMES(80), REFC(80), NFCSUP(80)
C
      COMMON /FRIC/ ALPHA,  BETA,  DELTA,  EPS
C
      COMMON /PRPS/ CPO,       GAMO,      DP(80),      SP(80),  RE(80),
     Z              CPC(80),   GAMC(80),  DUMR1(80),   DUMR2(80)
C
      COMMON /TCO/ ADUMP,      BTA,        CD,          CP,
     Z             GAM,        PIM,        R,           SPAN,        TOG,
     Z             WDUMP,      WIM,        AKC(15,80),  AKW(15,80),
     Z             A(400),     AJET(80),   AM2(80),     CNUM(80),
     Z             DH(80),     DHF(80),    DHJ(80),
     Z             DLX(400),   FF(80),     HC(80),      HG(80),
     Z             P(2,15,80),PEXIT(15),   PUMP(80),    QG(80),
     Z             QSNK(80),   RR(80),     S(15),       T(2,15,400),
     Z             TG(80),     TAU(400),   WFC(80),
     Z             WJ(15,80),  WCROS(2,15,80),          XN(80),
     Z             ICOR,       IFILM,      IHUB,        ITIP,
     Z             ISBLOK,     ISLICE,     NBLKSZ,      NSLICE,
     Z             NFWD,       NSTA,       IHC(80)
C
      COMMON /TRNSNT/ RHOC,      RHOM,      SPHTC,      SPHTM,
     Z                DLTYME,    TYME,      TEPS,       TYMMAX
C
      DATA CHKD/'@'/, UNCHKD/' '/
C
      DIMENSION DELTAN(15)
C
```

```
C FOLLOWING VARIABLES NEEDED TO CALCULATE FILM COOLING EFFECTIVENESS.
C  THEY ARE TRANSMITTED THROUGH COMMON FLMCOL
C WHERE-
C       PG IS GAS SIDE STATIC PRESSURE DISTRIBUTION;  XFC IS THE
C        DISTANCE A STATION IS DOWNSTREAM FROM THE NEAREST
C   ROW OF FILM COOLING HOLES, (IN);  FLMEFF IS THE CALCULATED
C       FILM EFFECTIVENESS AT EACH STATION;
C   XMUC IS COOLANT VISCOSITY BASED ON LOCAL COOLANT TEMPERATURE;
C   EMES IS THE PRODUCT M*S, WHERE M IS THE BLOWING
C   RATIO, AND S IS AN EQUIVALENT SLOT WIDTH;  REFC IS THE FILM
C   REYNOLDS NUMBER, BASED ON S;  AND NFCSUP IDENTIFIES
C   THE STATION NUMBER SUPPLYING FILM COOLING TO DOWNSTREAM STATIONS.
C
100   CONTINUE
C  INITIALIZE HOLE CHOKING INDICATOR TO UNCHOKED
      DO 101 I = 1,NSTA
101   WCHK(I) = UNCHKD
      ICHOKE = 0
      N = NSTA-1
C
C  N = NODE NUMBER OF LAST FLOW CHANNEL NODE, AT EXIT OF TRAILING EDGE
C
C
C  CALCULATE IMPINGEMENT FLOWS, AND FORWARD REGION FILM COOLING FLOWS
C
      TMP = TOG
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
      GAMO = GAM
      CPO = CP
      NODSF = 5*NFWD
C
      PAVG = P(2,ISLICE,1)
      WJ(ISLICE,1) = 0.0
      IF (PAVG.GT.PIM) GO TO 120
      WCR=CD*PAVG*AJET(1)/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*
     Z      (PIM/PAVG)**((GAMO-1.0)/GAMO)
      WJ(ISLICE,1)=PAVG/(R*TOG)*AJET(1)*CD*
     Z SQRT(64.4*GAMO*R*TOG/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO)))
     1  *(PIM/PAVG)**((GAMO-1.0)/GAMO)
      WFC(1) = 0.0
      IF (P(2,ISLICE,1).GT.PG(1)) WFC(1) = CD*.25*3.1415926*(DHF(1)**2)
     Z *SQRT(32.2*P(2,ISLICE,1)*(P(2,ISLICE,1)-PG(1))/(R*T(2,ISLICE,5)))
      IF(WCR.LT.WJ(ISLICE,1)) WCHK(1) = CHKD
      IF(WCR.LT.WJ(ISLICE,1)) WJ(ISLICE,1)=WCR
C
120   PAVG = P(2,ISLICE,2)
      WJ(ISLICE,2) = 0.0
      IF (PAVG.GT.PIM) GO TO 130
      WCR=CD*PAVG*AJET(2)/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*
     Z      (PIM/PAVG)**((GAMO-1.0)/GAMO)
      WJ(ISLICE,2)=PAVG/(R*TOG)*AJET(2)*CD*
     Z SQRT(64.4*GAMO*R*TOG/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO)))
     Z  *(PIM/PAVG)**((GAMO-1.0)/GAMO)
      WFC(2) = 0.0
      IF (P(2,ISLICE,2).GT.PG(2)) WFC(2) = CD*.25*3.1415926*(DHF(2)**2)
     Z *SQRT(32.2*P(2,ISLICE,2)*(P(2,ISLICE,2)-PG(2))/
     Z         (R*T(2,ISLICE,10)))
      IF(WCR.LT.WJ(ISLICE,2)) WCHK(2) = CHKD
      IF(WCR.LT.WJ(ISLICE,2)) WJ(ISLICE,2)=WCR
C
```

```
130   PAVG = P(2,ISLICE,3)
      WJ(ISLICE,3) = 0.0
      IF (PAVG.GT.PIM) GO TO 140
      WCR=CD*PAVG*AJET(3)/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*
     Z        (PIM/PAVG)**((GAMO-1.0)/GAMO)
      WJ(ISLICE,3)=PAVG/(R*TOG)*AJET(3)*CD*
     Z SQRT(64.4*GAMO*R*TOG/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO)))
     Z  *(PIM/PAVG)**((GAMO-1.0)/GAMO)
      WFC(3) = 0.0
      IF (P(2,ISLICE,3).GT.PG(3)) WFC(3) = CD*.25*3.1415926*(DHF(3)**2)
     Z *SQRT(32.2*P(2,ISLICE,3)*(P(2,ISLICE,3)-PG(3))/
     Z          (R*T(2,ISLICE,15)))
      IF(WCR.LT.WJ(ISLICE,3)) WCHK(3) = CHKD
      IF(WCR.LT.WJ(ISLICE,3)) WJ(ISLICE,3)=WCR
140   CONTINUE
C
      DO 150 I = 4,NFWD
      PAVG = P(2,ISLICE,I)
      WJ(ISLICE,I) = 0.0
      IF (PAVG.GT.PIM) GO TO 150
      WCR=CD*PAVG*AJET(I)/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*
     Z        (PIM/PAVG)**((GAMO-1.0)/GAMO)
      WJ(ISLICE,I)=PAVG/(R*TOG)*AJET(I)*CD*
     Z SQRT(64.4*GAMO*R*TOG/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO)))
     Z  *(PIM/PAVG)**((GAMO-1.0)/GAMO)
      WFC(I) = 0.0
      IF (P(2,ISLICE,I).GT.PG(I)) WFC(I) = CD*.25*3.1415926*(DHF(I)**2)
     Z *SQRT(32.2*P(2,ISLICE,I)*(P(2,ISLICE,I)-PG(I))/
     Z          (R*T(2,ISLICE,5*I)))
      IF(WCR.LT.WJ(ISLICE,I)) WCHK(I) = CHKD
      IF(WCR.LT.WJ(ISLICE,I)) WJ(ISLICE,I)=WCR
150   CONTINUE
C
C   CALCULATE CROSSFLOW RATE AT EACH STATION
C
200   WCROS(2,ISLICE,JS) = 0.0
      JDIS = JS/2
C**** JDIS IS THE DISPLACEMENT OF THE FLOW SPLIT STATION FROM STATION 1
      JSENS = JS - 2*JDIS
C********* JSENS = 0, STATION IS ON SUCTION SIDE (EVEN);
C                = 1,STATION IS ON PRESSURE SIDE (ODD)
      JP = JS + 2
      JM = JS - 2
      IF (JM.LT.1) JM = 1
      JFIN = JS + 1
      JSTART = JS + 3
      IF (JS.EQ.1) GO TO 220
      IF (JSENS.EQ.1) GO TO 230
      IF (JSENS.EQ.0) GO TO 270
C
220   CONTINUE
C***** WCROS AT A GIVEN STATION IS THE CROSSFLOW AT UPSTREAM STATION
C       PLUS IMPINGEMENT JET
C             FLOW FROM UPSTREAM STATION MINUS ANY FILM COOLING
C       FLOW AT THIS STATION.
C
C**  THIS BLOCK IS EXECUTED IF FLOW SPLIT OCCURS AT STATION 1.  ISTART=5
      WCROS(2,ISLICE,2) = DELTAN(ISLICE)*(WJ(ISLICE,1)-WFC(1)) - WFC(2)
      WCROS(2,ISLICE,3)=(1.-DELTAN(ISLICE))*(WJ(ISLICE,1)-WFC(1))-WFC(3)
      GO TO 320
```

```
C
230   CONTINUE
C*********  THIS BLOCK IS EXECUTED IF FLOW SPLIT STATION IS ON
C             THE PRESSURE SIDE.  JSENS=1
      WCROS(2,ISLICE,JP)=(1.-DELTAN(ISLICE))*(WJ(ISLICE,JS)-WFC(JS))
     Z-WFC(JP)
      WCROS(2,ISLICE,JM)=DELTAN(ISLICE)*(WJ(ISLICE,JS)-WFC(JS))-WFC(JM)
      IF (JM.EQ.1) GO TO 250
      IRNG = JDIS - 1
      DO 240 I = 1,IRNG
      IBK = (JM - 2*I)
      IBKP = IBK + 2
240   WCROS(2,ISLICE,IBK) = WCROS(2,ISLICE,IBKP) + WJ(ISLICE,IBKP)
     Z                           - WFC(IBK)
250   JNOD = JFIN
      DO 260 I = 2,JNOD,2
      IM8 = I-2
      IF (IM8.EQ.0) IM8 = 1
260   WCROS(2,ISLICE,I) = WCROS(2,ISLICE,IM8) + WJ(ISLICE,IM8)-WFC(I)
      GO TO 320
C
270   CONTINUE
C*********  THIS BLOCK IS EXECUTED IF FLOW SPLIT STATION IS ON
C           SUCTION SIDE.  JSENS=0
      WCROS(2,ISLICE,JP)=DELTAN(ISLICE)*(WJ(ISLICE,JS)-WFC(JS))-WFC(JP)
      IF (JS.EQ.2) JM = 1
      WCROS(2,ISLICE,JM)=(1.-DELTAN(ISLICE))*(WJ(ISLICE,JS)-WFC(JS))
     Z-WFC(JM)
      IF (JM.EQ.1) GO TO 300
      IF (JM.EQ.2) GO TO 290
      JM1 = JM-2
      DO 280 I = 2,JM1,2
      IBK = (JM-I)
      IBKP = IBK + 2
280   WCROS(2,ISLICE,IBK) = WCROS(2,ISLICE,IBKP) + WJ(ISLICE,IBKP)
     Z                           - WFC(IBK)
290   WCROS(2,ISLICE,1) = WCROS(2,ISLICE,2) + WJ(ISLICE,2) - WFC(1)
300   CONTINUE
C--------NOW UP THE PRESSURE SIDE
      JNOD = JFIN
      DO 310 I = 3,JNOD,2
310   WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) + WJ(ISLICE,I-2)-WFC(I)
C
320   CONTINUE
      ISTART = JSTART
      DO 330 I = ISTART,NFWD
      WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) + WJ(ISLICE,I-2) - WFC(I)
330   CONTINUE
C
C  CALCULATE CROSSFLOW RE & MACH NUMBER SQUARED, AND FILM COOLING RE,
C  FOR THE FORWARD REGION.
C
      DO 360 I = 1,NFWD
      LCOOL = 5*I
      LIN = LCOOL-1
      AMIN = A(LCOOL)
C
C    EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL
C      AND COOLANT BULK
C
```

```
      TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
      XMUC(I) = XMU
      CPC(I) = CP
      GAMC(I) = GAM
C
      RE(I) = 12.*3600.*ABS(WCROS(2,ISLICE,I))*DH(I)/(AMIN*XMU)
      REFC(I) = 12.*3600.*WFC(I)/(S(ISLICE)*XMU)
      IF (IHC(I).EQ.3) AMIN = A(LCOOL)*(SP(I)-DP(I))/SP(I)
      W = WCROS(2,ISLICE,I)
      PBAR = P(2,ISLICE,I)
      TBAR = T(2,ISLICE,LCOOL)
335   AM2(I) = (W/(PBAR*AMIN))**2*R*TBAR/GAMC(I)/32.2
      IF (AM2(I).LT.1.0) GO TO 340
      AMCHOK = AM2(I)
      ICHOKE = I
340   IF (IHC(I).NE.3) GO TO 350
      AM2(I) = AM2(I)*((AMIN/A(LCOOL))**2)
350   CONTINUE
360   CONTINUE
C
C  CALCULATE FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION.
C
      PAVG = .5*(P(2,ISLICE,NFWD) + P(2,ISLICE,NFWD-1))
      IF (PAVG.GT.PIM) GO TO 370
      WCR=CD*PAVG*ADUMP/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*
     Z      (PIM/PAVG)**((GAMO-1.0)/GAMO)
      WDUMP=PAVG/(R*TOG)*ADUMP*CD*
     Z SQRT(64.4*GAMO*R*TOG/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO)))
     Z  *(PIM/PAVG)**((GAMO-1.0)/GAMO)
      IF(WCR.LT.WDUMP) WCHKDM = CHKD
      IF(WCR.LT.WDUMP) WDUMP=WCR
C
C  ADD UP TOTAL FLOW FROM IMPINGEMENT PLENUM, WIM.
C
370   WIM = WDUMP
      DO 380 I = 1,NSTA
380   WIM = WIM + WJ(ISLICE,I)
C
C  TRAILING EDGE REGION, CALCULATE FILM COOLING FLOWS.
C
      ISTRT = NFWD+1
      DO 400 I = ISTRT,N,2
      LCOOL = 5*I
      WFCDUM = 0.0
390   IF (P(2,ISLICE,I).GT.PG(I)) WFCDUM = CD*.25*3.1415926*
     Z         SQRT(32.2*P(2,ISLICE,I)*(P(2,ISLICE,I)-PG(I))/
     Z              (R*T(2,ISLICE,LCOOL)))
      WFC(I) = WFCDUM*(DHF(I)**2)
      WFC(I+1) = WFCDUM*(DHF(I+1)**2)
400   CONTINUE
C
C  TRAILING EDGE REGION, CALCULATE CROSSFLOW, RE, MACH NUMBER
C  SQUARED, AND FILM COOLING RE.
C
      WCROS(2,ISLICE,ISTRT) = WCROS(2,ISLICE,NFWD-1) +
     Z               WCROS(2,ISLICE,NFWD) + WDUMP + WJ(ISLICE,NFWD-1)
     Z          + WJ(ISLICE,NFWD) - WFC(ISTRT) - WFC(ISTRT+1)
      WCROS(2,ISLICE,ISTRT+1) = WCROS(2,ISLICE,ISTRT)
      AMIN = A(NODSF+5)
```

```
C
C     EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL
C      AND COOLANT BULK
C
      TMP = (T(2,ISLICE,5*ISTRT) + T(2,ISLICE,5*ISTRT-1))/2.
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
      XMUC(ISTRT) = XMU
      XMUC(ISTRT+1) = XMU
      CPC(ISTRT) = CP
      CPC(ISTRT+1) = CP
      GAMC(ISTRT) = GAM
      GAMC(ISTRT+1) = GAM
C
      REFC(ISTRT) = 12.*3600.*WFC(ISTRT)/(S(ISLICE)*XMU)
      REFC(ISTRT+1) = 12.*3600.*WFC(ISTRT+1)/(S(ISLICE)*XMU)
      RE(ISTRT) = 12.*3600.*ABS(WCROS(2,ISLICE,ISTRT))*DH(ISTRT)/
     Z            (AMIN*XMU)
      RE(ISTRT+1) = RE(ISTRT)
      IS = ISTRT
      IF (IHC(IS).EQ.3) AMIN = A(NODSF+5)*(SP(IS)-DP(IS))/SP(IS)
      W = WCROS(2,ISLICE,ISTRT)
      PBAR = P(2,ISLICE,ISTRT)
      TBAR = T(2,ISLICE,NODSF+5)
405   AM2(ISTRT) = (W/(PBAR*AMIN))**2*R*TBAR/GAMC(ISTRT)/32.2
      IF (AM2(ISTRT).LT.1.0) GO TO 410
      AMCHOK = AM2(ISTRT)
      ICHOKE = ISTRT
410   IF (IHC(IS).NE.3) GO TO 420
      AM2(ISTRT) = AM2(ISTRT)*((AMIN/A(NODSF+5))**2)
420   AM2(ISTRT+1) = AM2(ISTRT)
      ISTRT = ISTRT + 2
      IS = NFWD + 1
      DO 450 I = ISTRT,N,2
      LCOOL = 5*I
      AMIN = A(LCOOL)
      IS = IS + 2
      IF (IHC(IS).EQ.3) AMIN = A(LCOOL)*(SP(IS)-DP(IS))/SP(IS)
      WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) - WFC(I) - WFC(I+1)
      WCROS(2,ISLICE,I+1) = WCROS(2,ISLICE,I)
C
C     EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL
C      AND COOLANT BULK
C
      TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LCOOL-1))/2.
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
      XMUC(I) = XMU
      XMUC(I+1) = XMU
      CPC(I) = CP
      CPC(I+1) = CP
      GAMC(I) = GAM
      GAMC(I+1) = GAM
C
      REFC(I) = 12.*3600.*WFC(I)/(S(ISLICE)*XMU)
      REFC(I+1) = 12.*3600.*WFC(I+1)/(S(ISLICE)*XMU)
      RE(I) = 12.*3600.*ABS(WCROS(2,ISLICE,I))*DH(I)/(A(LCOOL)*XMU)
      RE(I+1) = RE(I)
      W = WCROS(2,ISLICE,I)
      PBAR = P(2,ISLICE,I)
      TBAR = T(2,ISLICE,LCOOL)
425   AM2(I) = (W/(PBAR*AMIN))**2*R*TBAR/GAMC(I)/32.2
```

```
      IF (AM2(I).LT.1.0) GO TO 430                                      NFLOEST 0721
      AMCHOK = AM2(I)                                                   NFLOEST 0722
      ICHOKE = I                                                        NFLOEST 0723
430   IF (IHC(IS).NE.3) GO TO 440                                       NFLOEST 0724
      AM2(I) = AM2(I)*((AMIN/A(LCOOL))**2)                              NFLOEST 0725
440   AM2(I+1) = AM2(I)                                                 NFLOEST 0726
450   CONTINUE                                                          NFLOEST 0727
C                                                                       NFLOEST 0728
C  CALCULATE COOLANT CHANNEL FRICTION FACTOR AT EACH STATION.           NFLOEST 0729
C                                                                       NFLOEST 0730
      DO 560 I = 1,NSTA                                                 NFLOEST 0731
      LCOOL = 5*I                                                       NFLOEST 0732
      IF (WCROS(2,ISLICE,I).LE.0.0) GO TO 550                           NFLOEST 0733
C                                                                       NFLOEST 0734
C  DETERMINE IF RE IS LAMINAR, TRANSITIONAL, OR TURBULENT               NFLOEST 0735
C  AND CALCULATE THE FRICTION FACTOR                                    NFLOEST 0736
C                                                                       NFLOEST 0737
      IF (IHC(I).EQ.3) GO TO 540                                        NFLOEST 0738
      IF(RE(I).GT.2300.) GO TO 510                                      NFLOEST 0739
500   FF(I) = DELTA*RE(I)**EPS                                          NFLOEST 0740
      GO TO 560                                                         NFLOEST 0741
510   IF(RE(I).LT.4000.) GO TO 530                                      NFLOEST 0742
520   FF(I) = ALPHA*RE(I)**BETA                                         NFLOEST 0743
      GO TO 560                                                         NFLOEST 0744
530   A1=DELTA*2300.**EPS                                               NFLOEST 0745
      A2=ALPHA*4000.**BETA                                              NFLOEST 0746
      FF(I) = (A2*(RE(I)-2300.)+A1*(4000.-RE(I)))/1700.                 NFLOEST 0747
      GO TO 560                                                         NFLOEST 0748
540   CONTINUE                                                          NFLOEST 0749
C                                                                       NFLOEST 0750
C  FOR A PIN FIN ARRAY:                                                 NFLOEST 0751
      FF(I) = 1.060*(RE(I)**(-.3301))                                   NFLOEST 0752
      GO TO 560                                                         NFLOEST 0753
550   CONTINUE                                                          NFLOEST 0754
      FF(I) = 0.0                                                       NFLOEST 0755
560   CONTINUE                                                          NFLOEST 0756
C                                                                       NFLOEST 0757
C                                                                       NFLOEST 0758
C                                                                       NFLOEST 0759
C  THE FOLLOWING BLOCK IS USED TO COMPUTE THE FILM COOLING EFFECTIVENESSNFLOEST 0760
C     IF IFILM IS SET = 2                                               NFLOEST 0761
C                                                                       NFLOEST 0762
      IF (IFILM.LT.2) GO TO 690                                         NFLOEST 0763
      IFSPLT = 0                                                        NFLOEST 0764
      DO 610 I = 1,NSTA                                                 NFLOEST 0765
610   XFC(I) = 0.0                                                      NFLOEST 0766
      N = NSTA-1                                                        NFLOEST 0767
C LOCATE FILM COOLING HOLES AND SET UP THE XFC ARRAY                    NFLOEST 0768
C                                                                       NFLOEST 0769
C---IFSPLT IS AN INDICATOR THAT TELLS WHICH SIDE OF THE BLADE STATION 1 NFLOEST 0770
C---         IS TO BE CONSIDERED A PART OF FOR FILM COOLING PURPOSES.   NFLOEST 0771
C---        = 0 IS THE DEFAULT, AND INDICATES SUCTION SIDE              NFLOEST 0772
C---        = 1 WILL INDICATE PRESSURE SIDE.                            NFLOEST 0773
C FIRST, MARCH DOWN THE PRESSURE SIDE SEARCHING FOR FILM COOLING HOLES  NFLOEST 0774
      IF (DHF(1).GT.0.0) NFC = IFSPLT                                   NFLOEST 0775
      IF (NFC.EQ.0) NFC = NSTA + 1                                      NFLOEST 0776
      XDUM = 0.0                                                        NFLOEST 0777
      DO 615 I = 3,NSTA,2                                               NFLOEST 0778
      NOS = 5*I - 4                                                     NFLOEST 0779
      IF (I.GT.NFC) XFC(I) = XDUM + DLX(NOS)                            NFLOEST 0780
```

```
      IF (DHF(I).GT.0.0) GO TO 612
      XDUM = XFC(I)
      GO TO 615
612   NFC = I
      XDUM = 0.0
615   CONTINUE
C
C-- SUCTION SIDE
C
      IF (DHF(1).GT.0.0) NFC = 1 - IFSPLT
      IF (NFC.EQ.0) NFC = NSTA + 1
      XDUM = 0.0
      DO 625 I = 2,N,2
      NOS = 5*I - 4
      IF (I.GT.NFC) XFC(I) = XDUM + DLX(NOS)
      IF (DHF(I).GT.0.0) GO TO 622
      XDUM = XFC(I)
      GO TO 625
622   NFC = I
      XDUM = 0.0
625   CONTINUE
C INT. J. HT. & MASS TRANS., V8, 1965, PP 55-65
C    FLMEFF = 3.09*((X/(M*S))*(RE*MUC/MUG)**(-1/4)  + 4.1)**(-.8)
C
      IFCS = 0
      IFCP = 0
      IF (WFC(1).LE.0.0) GO TO 630
      IFCS = 1 - IFSPLT
      IFCP = IFSPLT
      IF (RHOVGA(1).GT.0.0) EMES(1) = 144.*WFC(1)/(RHOVGA(1)*S(ISLICE))
630   CONTINUE
      DO 650 I=2,NSTA
      ISENS = I - 2*(I/2)
      IF (RHOVGA(I).GT.0.) EMES(I) = 144.*WFC(I)/(RHOVGA(I)*S(ISLICE))
      FLMEFF(I) = 0.0
      IF (ISENS.EQ.0) GO TO 640
C PRESSURE SIDE SUPPLY HOLE LOCATIONS
      IF (WFC(I).GT.0.0) IFCP = I
      NFCSUP(I) = IFCP
      GO TO 650
640   CONTINUE
C SUCTION SIDE SUPPLY HOLE LOCATIONS
      IF (WFC(I).GT.0.) IFCS = I
      NFCSUP(I) = IFCS
650   CONTINUE
C
      TMP = TG(1)
      CALL GASTBL(TMP,C,CPM,GAM,PD,R,XMUM)
C
C  FINALLY, CALCULATE THE EFFECTIVENESS
C
      DO 680 I = 1,NSTA
      IMS = NFCSUP(I)
      IF (XFC(I).EQ.0.0.OR.EMES(IMS).EQ.0.0.OR.REFC(IMS).EQ.0.0)
     Z                                                     GO TO 680
      C3 = CPC(IMS)/CPM
      XBAR = (XFC(I)/EMES(IMS))*((REFC(IMS)*XMUC(I)/XMUM)**(-.25))
      ETAPRM = 3.09*(XBAR+4.1)**(-.8)
      FLMEFF(I) = C3*ETAPRM/(1.0 + (C3-1.0)*ETAPRM)
      IF (FLMEFF(I).GT.1.0) FLMEFF(I) = 1.0
```

```
680   CONTINUE
C
C
690   CONTINUE
      RETURN
      END


C----SOURCE.NFLSPLP
      SUBROUTINE FLSPLT(AJET,EPSN,ISLICE,NODSF,IDELT,JS,DELTAN,ICONV)
      DIMENSION DELTAN(15), AJET(80), JSOLDS(25)
C
C-   SOURCE.NFLSPLP---A SUBROUTINE TO SET THE STATION AT WHICH COOLING
C     AIR FLOW SPLITS BETWEEN THE SUCTION
C            AND THE PRESSURE SIDE FLOW CHANNELS.
C
C    INPUT TO FLSPLT IS THE PRESSURE MATCH PARAMETER, EPSN; THE NO. OF
C     NODES (NODSF) IN THE IMPINGEMENT REGION;
C          JS COMES IN AS THE CURRENT FLOW SPLIT STATION NO., AND IS
C          RETURNED AS THE NEW STATION IF A CHANGE IS NEEDED.
C          DELTAN COMES IN AS THE CURRENT FRACTION OF FLOW SPLIT TO
C           SUCTION SIDE FROM AN IMPINGEMENT
C             JET AT JS. IF A CHANGE IN JS IS NOT NEEDED, DELTAN IS
C             USED TO FINE TUNE THE SPLIT.
C          ICONV INDICATES IF CONVERGENCE IS COMPLETE.
C             = 0--NOT DONE; = 1--OK.
C
      NFWD = NODSF/5
      IF (IUNSTB.EQ.1) GO TO 280
      IF (IDELT.NE.1) GO TO 220
      JNUMS = 0
      IUNSTB = 0
      NUMS = 0
      JSGNCH=0
      JOUTRG=0
      DO 210 I = 1,25
210   JSOLDS(I) = 0
220   CONTINUE
      CRITR = .002
      ICONV = 0
      JSENS = JS - 2*(JS/2)
C********  (SUCTION   - PRESSURE SIDE PRESSURES)/ SUCTION SIDE  = EPSN
      IF (ABS(EPSN).LT.CRITR) GO TO 280
C
C***********IF EPSN < 0.0; NEED TO INCREASE FLOW TO PRESSURE SIDE
C***********   EPSN > 0.0; NEED TO INCREASE FLOW TO SUCTION SIDE
C
      IF (JTIMES.EQ.0) GO TO 246
C
C********* JTIMES = 0, THIS IS FIRST CHECK AT THIS STATION,
C                     SO ROUGH ADJUST DELTAN;
C*********          1, HAVE BEEN HERE BEFORE, SO FINE TUNE DELTAN.
C
      IF ( JSGNCH.GT.0 ) GO TO 247
C
C********* JSGNCH = 0, THERE HAS NOT BEEN A PRIOR SIGN CHANGE IN EPSN;
C*********        = 1, THERE HAS BEEN A SIGN CHANGE BEFORE,
C                       SO STAY AT THIS STATION
C
C
C
242   IF (EPSO/EPSN.LT.0.) JSGNCH = 1
```

```
C
 247   CONTINUE
       IF (EPLAST/EPSN.GE.0) GO TO 243
       DELTAO = DELAST
       EPSO = EPLAST
 243   CONTINUE
       IF (JSGNCH.EQ.0) GO TO 252
       IF (NUMS.GT.0) GO TO 248
       EPSMIN = ABS(EPSN)
       DLTAOP = DELTAN(ISLICE)
 248   NUMS = NUMS + 1
       IF (ABS(EPSN).GT.EPSMIN) GO TO 249
       EPSMIN = ABS(EPSN)
       DLTAOP = DELTAN(ISLICE)
 249   CONTINUE
       IF (NUMS.LT.4) GO TO 252
       IF (JNUMS.EQ.1) GO TO 250
       NUMS = 0
       JNUMS = 1
       DELTAN(ISLICE) = DELTAO
       JTIMES = 0
       JSGNCH = 0
       JOUTRG = 0
       GO TO 290
 250   CONTINUE
       DELAST = DELTAN(ISLICE)
       DELTAN(ISLICE) = DLTAOP
       IUNSTB = 1
       GO TO 290
C
C
 246   JTIMES = 1
       EPSO = EPSN
       DELTAO = DELTAN(ISLICE)
       IF (EPSO.GT.0.0) DELTAN(ISLICE) = (1.0+DELTAN(ISLICE))/2.0
       IF (EPSO.LT.0.0) DELTAN(ISLICE) = DELTAN(ISLICE)/2.
       IF (DELTAN(ISLICE).EQ.DELTAO) DELTAN(ISLICE) = DELTAN(ISLICE)+
      Z                                   (.5-DELTAN(ISLICE))/5.
       GO TO 290
C
C
 252   CONTINUE
       TERM = EPSN*(DELTAO-DELTAN(ISLICE))/(EPSO-EPSN)
       IF (TERM.EQ.0.) TERM = .05
       IF (JSGNCH.GT.0) GO TO 255
       DELTAO = DELTAN(ISLICE)
       EPSO = EPSN
 255   CONTINUE
       DELAST = DELTAN(ISLICE)
       DELTAN(ISLICE) = DELTAN(ISLICE)-TERM
       IF ( DELTAN(ISLICE).LT.1.0.AND.DELTAN(ISLICE).GT.0.0) GO TO 290
       IF (JOUTRG.GT.0) GO TO 258
       IF (DELTAN(ISLICE).LT.0.0) DELTAN(ISLICE) = .01
       IF (DELTAN(ISLICE).GT.1.0) DELTAN(ISLICE) = .99
       JOUTRG = 1
       GO TO 290
C
C
 258   CONTINUE
       JOUTRG = 0
```

```
C
C
      JSGNCH = 0
      JTIMES = 0
      JSOLDS(JS) = 1
      IF (DELTAN(ISLICE).LT.1.) GO TO 265
C
C*** MOVE JS IN PRESSURE DIRECTION
      IF (JSENS.EQ.0) GO TO 262
261   JS = JS + 2
      IF (AJET(JS).LE.0.) GO TO 261
      GO TO 285
C
C
262   CONTINUE
      IF (JS.EQ.2) JS = 1
      IF (JS.GT.2) JS = JS - 2
      IF (AJET(JS).LE.0.) GO TO 262
      GO TO 285
C
C
265   CONTINUE
C*** MOVE JS IN SUCTION DIRECTION
      IF (JSENS.EQ.0) GO TO 267
      IF (JS.EQ.1) JS = 2
      IF (JS.GE.3) JS = JS - 2
      IF (AJET(JS).GT.0.) GO TO 285
      JSENS = JS - 2*(JS/2)
      GO TO 265
C
C
267   CONTINUE
      JS = JS + 2
      IF (AJET(JS).LE.0.) GO TO 267
      GO TO 285
C
C********* GET READY TO LEAVE SUBROUTINE
C
C THIS BLOCK IS EXECUTED IF CONVERGENCE WAS DETECTED
C
280   ICONV = 1
      JTIMES = 0
      IF (IUNSTB.EQ.1) WRITE(6,284) ISLICE,IDELT,JS,DELTAN(ISLICE)
      IF (IUNSTB.EQ.1) WRITE(8,284) ISLICE,IDELT,JS,DELTAN(ISLICE)
      IUNSTB = 0
      EPLAST = EPSN
      RETURN
284   FORMAT(1H2,40(' '),40('*')//' SLICE ',I2,', POOR FLOW SPLIT, ',
     Z        I3,' ITERATIONS, SPLIT AT STATION ',
     Z I2,', BEST SPLIT IS AT DELTA = ',F6.4)
C
C
C THIS BLOCK IS EXECUTED FOR AN ABNORMAL EXIT---PROGRAM IS TERMINATED
C
789   CONTINUE
      WRITE(6,792) DELTAN(ISLICE)
      WRITE(8,792) DELTAN(ISLICE)
      STOP
792   FORMAT(/5X,' ****** FLOW SPLIT CANNOT BE MADE AS SPECIFIED,',
     Z        ' DELTA = ',F9.5)
```

```
C
C
285   IF ( JSOLDS(JS).EQ.1) GO TO 789
      DELTAN(ISLICE) = .50
C
C
C THIS BLOCK IS THE USUAL EXIT AFTER ADJUSTING THE FLOW SPLIT
C
290   CONTINUE
      EPLAST = EPSN
      IF (JSGNCH.EQ.0) DELAST = DELTAO
      IDELT = IDELT + 1
      RETURN
      END


C----SOURCE.NGASDAT
      BLOCK DATA
C
C--  SOURCE.NGASDAT---
C
      COMMON /GAAS/ GS(200),NG
      DATA GS/620.,    1160.,    1700.,    2240.,    2780.,    3320.,
     Z     .02564,  .03580,  .04548,  .05467,  .06435,  .07475,
     Z      .2511,   .2681,   .2814,   .2939,   .3070,   .3214,
     Z       .706,    .706,    .705,    .703,    .702,    .699,
     Z     .07233,  .09458,  .11369,  .13063,  .14683,  .16256,
     Z       170*0.0/, NG /6/
C
C---GS IS TABLE OF AIR PROPERTIES VS TEMPERATURE AT CONSTANT PRESSURE
C---   PROPERTY VALUES ARE FROM POFERL & SVEHLA, TN D-7488, AT 20 ATM.
C---NG IS THE NUMBER OF TEMPERATURE ENTRIES IN THE TABLE
C--- ENTRIES IN GS ARE:
C---     1ST NG ARE TEMPERATURE,(F)
C---     2ND NG ARE THERMAL CONDUCTIVITY, (BTU/(HR*FT*R))
C---     3RD NG ARE SPECIFIC HEAT, (BTU/(LBM*R))
C---     4TH NG ARE PRANDTL NUMBER
C---     5TH NG ARE VISCOSITY, (LBM/(FT*HR))
C
      END
C----SOURCE.NGASTB
      SUBROUTINE GASTBL(TMP,C,CP,GAM,PD,R,XMU)
C
C-   SOURCE.NGASTB
C
C  A SUBROUTINE TO LOOK UP GAS PROPERTIES IN AN INPUT TABLE (GS(200))
C   WHERE  TMP = TEMPERATURE AT WHICH PROPERTIES ARE TO BE EVALUATED (R)
C          C   = GAS THERMAL CONDUCTIVITY (BTU/(HR*FT*R))
C          CP  = GAS SPECIFIC HEAT (BTU/(LBM*R))
C          GAM = RATIO OF SPECIFIC HEATS
C          PD  = PRANDTL NUMBER
C          R   = SPECIFIC GAS CONSTANT (FT*LBF)/(LBM*R)
C          XMU = VISCOSITY (LBM/(FT*HR))
C
      COMMON /GAAS/ GS(200),NG
      DIMENSION AC(5)
C
      TMP1=TMP - 460.
      IF(TMP1.GT.GS(1)) GO TO 200
100   AP1=0.0
      AP2=1.0
      I1=2
```

```
      I2=1
      GO TO 500
200   DO 300 I=1,NG
      I1=I
      IF(GS(I).GT.TMP1) GO TO 400
300   CONTINUE
      TMP1=GS(NG)
400   I2=I1-1
      AP1=(TMP1-GS(I2))/(GS(I1)-GS(I2))
      AP2=1.0-AP1
500   DO 600 J=1,4
      I1=I1+NG
      I2=I2+NG
600   AC(J)=AP1*GS(I1)+AP2*GS(I2)
      AC(5)=1.0/(1.0-R/(778.2*AC(2)))
      C=AC(1)
      CP=AC(2)
      PD=AC(3)
      XMU=AC(4)
      GAM=AC(5)
      RETURN
      END
```

```
C----SOURCE.NGAUS
      SUBROUTINE GAUSS(N,K)
C
C-   SOURCE.NGAUS----
C
C-----GIVEN A COMPRESSED VERSION OF AN AUGMENTED, BAND MATRIX A
C-----WERE K IS THE WIDTH OF THE BAND, N IS THE NUMBER OF ROWS (MAX 400)
C-----DIAGONAL ELEMENTS OF THE ORIGINAL MATRIX ARE STORED IN
C----- COLUMN ((K/2)+1) -- TCOF(I,((K/2)+1))
C-----THE ORIGINAL RIGHT HAND SIDE IS IN COLUMN K+1--- TCOF(I,K+1)
C-----GAUSS ELIMINATION IS USED TO MAKE ALL ELEMENTS BELOW
C----- THE DIAGONAL ZERO.
C-----BACK-SUBSTITUTION IS USED TO COMPUTE THE X'S, WHICH ARE
C----- RETURNED IN TCOF(I,K+1)
C
      REAL*8 TCOF
      COMMON /MATRX/ TCOF(400,30)
      IWR = 0
      NROW = 0
      IF (IWR.EQ.0) GO TO 63
C
C  DEBUGGING OUTPUT:
C
C  IWR CAN BE SET DYNAMICALLY IN ORDER TO GET DEBUG OUTPUT OF
C  SELECTED ROWS OF THE MATRIX, BEFORE OR AFTER REDUCTION.
C
      WRITE(8,57)
57    FORMAT(' ENTER NUMBER OF ROW TO BE DISPLAYED. USE I3 FORMAT')
58    READ(7,59) NROW
59    FORMAT(I3)
      IF (NROW.EQ.0) GO TO 63
      KP = K+1
      WRITE(8,60) NROW
      WRITE(8,61) (I,TCOF(NROW,I),I=1,KP)
60    FORMAT(/'  TCOF MATRIX, ROW NO. ',I3)
61    FORMAT(5(' (',I3,',',D17.10,')'))
      WRITE(8,62)
62    FORMAT(/' ENTER ANOTHER ROW NO. OR 000 TO CONTINUE PROCESSING')
```

```
      GO TO 58
C
C
63    CONTINUE
65    JPIV = K/2 + 1
      N1 = N-1
      DO 100 I = 1,N1
      JS = I+1
      JF = I + K/2
      IF (JF.GT.N) JF = N
      PIVOT = TCOF(I,JPIV)
      IF (PIVOT.EQ.0.0) GO TO 130
      DO 90 J = JS,JF
      JR = JPIV-J+I
      IF (DABS(TCOF(J,JR)).LT.1.0D-30) GO TO 90
      FM = TCOF(J,JR)/PIVOT
      TCOF(J,JR) = 0.0
      LS = JR + 1
      LF = LS + K/2
      IF (LF.LT.LS) GO TO 85
      DO 80 L = LS,LF
      LR = L+JPIV+1-LS
      IF (LR.GT.K) GO TO 85
80    TCOF(J,L) = TCOF(J,L) - FM*TCOF(I,LR)
85    TCOF(J,K+1) = TCOF(J,K+1) - FM*TCOF(I,K+1)
90    CONTINUE
100   CONTINUE
C
C
155   IF (IWR.EQ.0) GO TO 163
C
C  DEBUGGING OUTPUT:
C
      WRITE(8,57)
158   READ(7,59) NROW
      IF (NROW.EQ.0) GO TO 163
      KP = K+1
      WRITE(8,60) NROW
      WRITE(8,61) (I,TCOF(NROW,I),I=1,KP)
      WRITE(8,62)
      GO TO 158
C
C
163   CONTINUE
      TCOF(N,K+1) = TCOF(N,K+1)/TCOF(N,JPIV)
      DO 120 I = 1,N1
      IIN = N-I
      JF = K/2
      SUM = TCOF(IIN,K+1)
      DO 115 J = 1,JF
      JP = J + JPIV
      IJ = J+IIN
      IF (IJ.GT.N) GO TO 117
115   SUM = SUM - TCOF(IJ,K+1)*TCOF(IIN,JP)
117   CONTINUE
120   TCOF(IIN,K+1) = SUM/TCOF(IIN,JPIV)
125   CONTINUE
      RETURN
130   WRITE(7,135) I
135   FORMAT(/'  DIAGONAL ELEMENT FOR ROW ',I2,' IS ZERO. NO ',
```

```
     Z        'FURTHER ATTEMPT TO SOLVE WILL BE MADE.')                NGAUS    1201
      GO TO 125                                                        NGAUS    1202
      END                                                              NGAUS    1203


C----SOURCE.NGETINT                                                    NGETINT 1204
      SUBROUTINE GETIN                                                 NGETINT 1205
C                                                                      NGETINT 1206
C-   SOURCE.NGETINT----                                                NGETINT 1207
C                                                                      NGETINT 1208
      COMMON /BOUND/ BCXS(400),   BCXP(400),   BCHGS(1000), BCHGP(1000),NGETINT 1209
     Z               BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),NGETINT 1210
     Z               BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),  NGETINT 1211
     Z               QHUBIN(400), QHUB(80),    TTIPIN(400), TTIP(80),  NGETINT 1212
     Z               QTIPIN(400), QTIP(80),    RHOVG(400),  PEX(400),  NGETINT 1213
     Z               BCTIME(50),  TTIO(50),    PTIO(50),    WPLEN,     NGETINT 1214
     Z               WSVST(50),   AKCTBL(20),  AKWTBL(20),  NBCS,  NBCP NGETINT 1215
C                                                                      NGETINT 1216
      COMMON /FLMCOL/ RHOVGA(80), PG(80),    XFC(80),  FLMEFF(80),     NGETINT 1217
     Z                XMUC(80),   EMES(80), REFC(80), NFCSUP(80)       NGETINT 1218
C                                                                      NGETINT 1219
      COMMON /IMPCOR/ CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, CIMP6, CIMP7, NGETINT 1220
     Z                DIMP1, DIMP2, DIMP3, DIMP4, DIMP5, DIMP6         NGETINT 1221
C                                                                      NGETINT 1222
      COMMON /RADL/ APLN(15),  DPLN(15),   RIN(15),    ROUT(15),       NGETINT 1223
     Z              PIN(15),   TIN(15),    W(15),      WS              NGETINT 1224
C                                                                      NGETINT 1225
      COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),             NGETINT 1226
     Z                IPLOT,  MD1,  MD2,  MD3, IADJIN, IWRITE          NGETINT 1227
C                                                                      NGETINT 1228
      COMMON /TCO/ ADUMP,     BTA,       CD,         CP,               NGETINT 1229
     Z             GAM,       PIM,       R,          SPAN,       TOG,  NGETINT 1230
     Z             WDUMP,     WIM,       AKC(15,80), AKW(15,80),       NGETINT 1231
     Z             A(400),    AJET(80),  AM2(80),    CNUM(80),         NGETINT 1232
     Z             DH(80),    DHF(80),   DHJ(80),                      NGETINT 1233
     Z             DLX(400),  FF(80),    HC(80),     HG(80),           NGETINT 1234
     Z             P(2,15,80),PEXIT(15), PUMP(80),   QG(80),           NGETINT 1235
     Z             QSNK(80),  RR(80),    S(15),      T(2,15,400),      NGETINT 1236
     Z             TG(80),    TAU(400),  WFC(80),                      NGETINT 1237
     Z             WJ(15,80), WCROS(2,15,80),        XN(80),           NGETINT 1238
     Z             ICOR,      IFILM,     IHUB,       ITIP,             NGETINT 1239
     Z             ISBLOK,    ISLICE,    NBLKSZ,     NSLICE,           NGETINT 1240
     Z             NFWD,      NSTA,      IHC(80)                       NGETINT 1241
C                                                                      NGETINT 1242
      COMMON /TRNSNT/ RHOC,      RHOM,      SPHTC,      SPHTM,         NGETINT 1243
     Z                DLTYME,    TYME,      TEPS,       TYMMAX         NGETINT 1244
C                                                                      NGETINT 1245
      COMMON /UNITS/ CINCH(2), CHTC(2),  CHFLX(2), CPRSR(2), CMSFL(2), NGETINT 1246
     Z               CTMPF(2), CTCON(2), CDEN(2),  CSPHT(2), CGASC(2), NGETINT 1247
     Z               CVISC(2), CRHOVG(2),IUNITS                        NGETINT 1248
C                                                                      NGETINT 1249
      DIMENSION      THK(3),    TDLX(5),   TFLMHL(10)                  NGETINT 1250
C                                                                      NGETINT 1251
      NAMELIST /TITL/ TITLE                                            NGETINT 1252
C                                                                      NGETINT 1253
      NAMELIST /CHANLS/ NSLICE, NSTA,   INEDIT, IPLOT,  IWRITE,        NGETINT 1254
     Z                  MD1,    MD2,    MD3,    IUNITS, IFILM,  IADJIN NGETINT 1255
C                                                                      NGETINT 1256
      NAMELIST /BC/     NBCS,   NBCP,   BCXS,   BCXP,   BCHGS,  BCHGP, NGETINT 1257
     Z                  BCTGS,  BCTGP,  BCQGS,  BCQGP,  BCPGS,  BCPGP, NGETINT 1258
     Z                  THUBIN, QHUBIN, TTIPIN, QTIPIN, RHOVG,         NGETINT 1259
     Z                  PEX,    BCTIME, TTIO,   PTIO,   WPLEN,         NGETINT 1260
```

```
     Z                   WSVST,  AKCTBL, AKWTBL, RHOC,    RHOM,             NGETINT 1261
     Z                   SPHTC,  SPHTM,  DLTYME, TEPS,    TYMMAX            NGETINT 1262
C                                                                           NGETINT 1263
      NAMELIST /CONTRL/ NFWD,    ICOR,   NGEO                               NGETINT 1264
C                                                                           NGETINT 1265
      NAMELIST /PROPS/  CD,      SPAN,   ADUMP,  DHYD,    APLEN,   RO,      NGETINT 1266
     Z                  RI,      CIMP1,  CIMP2,  CIMP3,   CIMP4,   CIMP5,   NGETINT 1267
     Z                  CIMP6,   CIMP7,  DIMP1,  DIMP2,   DIMP3,   DIMP4,   NGETINT 1268
     Z                DIMP5,  DIMP6                                         NGETINT 1269
C                                                                           NGETINT 1270
      NAMELIST /GEO/    ISTA,    ISTB,   THK,    TDLX,    TDHJ,    TXN,     NGETINT 1271
     Z                  TDHF,    TRR,    IHCT,   TDP,     TSP,     TFLMHL   NGETINT 1272
C                                                                           NGETINT 1273
      DATA TIKLE/'    '/                                                    NGETINT 1274
C                                                                           NGETINT 1275
C  NSLICE = THE NO. OF SLICES OF THE BLADE THAT ARE BEING CONSIDERED        NGETINT 1276
C  IHUB   = 1 INDICATES A SPECIFIED TEMPERATURE DISTRIBUTION IS GIVEN ATNGETINT 1277
C           THE HUB END (F)                                                 NGETINT 1278
C         = 2 INDICATES AN ADIABATIC SURFACE AT THE HUB END                 NGETINT 1279
C         = 3 INDICATES HEAT FLUX IS SPECIFIED AT HUB END                   NGETINT 1280
C           (BTU/HR FT**2 R)                                                NGETINT 1281
C  ITIP   = 1 INDICATES A SPECIFIED TEMPERATURE DISTRIBUTION IS GIVEN ATNGETINT 1282
C           THE TIP END (F)                                                 NGETINT 1283
C         = 2 INDICATES AN ADIABATIC SURFACE AT THE TIP END                 NGETINT 1284
C         = 3 INDICATES HEAT FLUX IS SPECIFIED AT TIP END                   NGETINT 1285
C            (BTU/HR FT**2 R)                                               NGETINT 1286
C  IADJIN = 0, MEANS TO HOLD PTIO CONSTANT AND ADJUST WPLEN;                NGETINT 1287
C         > 0, MEANS TO FIX WPLEN AND ADJUST PTIO.                          NGETINT 1288
C  ISTA = FIRST STATION NUMBER FOR THIS DATA SET                            NGETINT 1289
C  IF ISTB IS SPECIFIED, IT IS THE LAST STATION NUMBER FOR THIS DATA SETNGETINT 1290
C  IF ISTB IS SPECIFIED, IT MUST BE EQUL TO ISTA + A MULTIPLE OF 2.         NGETINT 1291
C  THK =   (1)-COATING THICKNESS, (2)-METAL THICKNESS, AND (3)-CHANNEL      NGETINT 1292
C         WIDTH. ALL IN INCHES.                                             NGETINT 1293
C  TDLX = DISTANCE FROM UPSTREAM NODE (INCHES)                              NGETINT 1294
C  TDHJ = HYDRAULIC DIAMETER OF IMPINGEMENT JET HOLE (INCHES) - STORED      NGETINT 1295
C         UNDER STATION NUMBER                                              NGETINT 1296
C  TDHF = EFFECTIVE DIAMETER OF FILM COOLING HOLE IF PRESENT (INCHES) -     NGETINT 1297
C         STORED UNDER STATION NUMBER                                       NGETINT 1298
C       = DIAMETER OF ONE HOLE*SQRT(NO. OF HOLES AT THIS STATION IN THISNGETINT 1299
C          SLICE)                                                           NGETINT 1300
C  TXN = SPANWISE SPACING OF IMPINGEMENT JETS (INCHES)                      NGETINT 1301
C  TRR = RADIAL LOCATION OF THIS STATION (INCHES)                           NGETINT 1302
C  IHCT  INDICATES THE TYPE OF INSIDE HEAT TRANSFER AT THIS STATION,        NGETINT 1303
C       = 1  FOR IMPINGEMENT WITH CROSSFLOW                                 NGETINT 1304
C       = 2  FOR FORCED CONVECTION CHANNEL FLOW                             NGETINT 1305
C       = 3  FOR PIN FINS                                                   NGETINT 1306
C  TDP  = THE PIN FIN DIAMETER (IN) IF PINS ARE USED;                       NGETINT 1307
C  TSP  = THE PIN FIN SPACING (IN), ASSUMING AN EQUILATERAL TRIANGULAR      NGETINT 1308
C         ARRAY OF PINS.                                                    NGETINT 1309
C  AKCTBL= TABLE OF CLADDING THERMAL CONDUCTIVITY (BTU/HR FT R) VS          NGETINT 1310
C         TEMPERATURE (F)                                                   NGETINT 1311
C  AKWTBL= TABLE OF WALL METAL THERMAL CONDUCTIVITY (BTU/HR FT R) VS        NGETINT 1312
C         TEMPERATURE (F)                                                   NGETINT 1313
C  RHOVG = HOT GAS FREE STREAM MASS VELOCITY, DENSITY*VELOCITY, FOR FILMNGETINT 1314
C          COOLING USE, AT EACH FILM COOLING STATION.                       NGETINT 1315
C          INPUT IN (LBM/SEC FT**2),  OR (KG/SEC M**2) IF IUNITS=1          NGETINT 1316
C  RHOC  = DENSITY OF OUTER COATING (LBM/FT**3)                             NGETINT 1317
C  RHOM  = DENSITY OF WALL METAL (LBM/FT**3)                                NGETINT 1318
C  SPHTC = SPECIFIC HEAT OF COATING (BTU/LBM R)                             NGETINT 1319
C  SPHTM = SPECIFIC HEAT OF WALL METAL                                      NGETINT 1320
```

```
C  DLTYME = TIME STEP SIZE FOR TRANSIENT CALCULATIONS (SEC)
C  TYMMAX = MAX. TIME (SEC) TO WHICH TRANSIENT IS CARRIED.
C  TEPS   = FRACTION OF TIME STEP AT WHICH TEMP. IS EVALUATED.( NEW = OLD
C          + TEPS*(NEW-OLD))
C  WSVST  = TABLE OF WHEEL SPEED VS TIME, (RPM VS SEC), ODD SUBSCRIPTS
C          ARE SPEED,EVEN ARE TIME, WSVST(2)=0.0
C
C  CIMP1 TO CIMP5 ARE EXPONENTS TO BE USED IN A GENERAL
C     IMPINGEMENT WITH CROSSFLOW CORRELATION. IF NOT SPECIFIED, THEN THE
C     BUILT IN KIRCHER-TABAKOFF CORRELATION IS USED.
C   SEE SUBROUTINE HCOOLT FOR DESCRIPTION OF GENERAL CORRELATION.
C
C  INITIALIZE:
C
100   CONTINUE
      IEND = 0
      IADJIN = 0
      IHUB = 2
      ITIP = 2
      CIMP1 = 0.0
      DIMP1 = 0.0
      ADUMP = 0.0
      IFILM = 0
      IUNITS = 2
      ALPHA = .04
      BETA = -.16
      DELTA = 16.
      EPS = -1.
      CD = .8
C
C  GAS CONSTANT FOR AIR, FT LBF/LBM R
      R = 53.35
C
C-- SET VALUES FOR UNITS CORRECTION FACTORS---
C-- ...(1) CONVERTS FROM SI TO ENGLISH, ...(2) MAKES NO CONVERSION--
C          ALREADY IN ENGLISH
C
C--- CINCH(1) IS CONVERSION FACTOR FROM (CM) TO (IN)
      CINCH(1) = .39370
      CINCH(2) = 1.0
C--- CHTC(1) IS CONVERSION FACTOR FROM (WATTS/M**2 K) TO (BTU/HR FT**2R)
      CHTC(1) = .17623
      CHTC(2) = 1.0
C--- CHFLX(1) IS CONVERSION FACTOR FROM (WATTS/M**2) TO (BTU/HR FT**2)
      CHFLX(1) = .31721
      CHFLX(2) = 1.0
C--- CPRSR(1) IS CONVERSION FACTOR FROM (KILOPASCALS) TO (PSIA)
      CPRSR(1) = .14503
      CPRSR(2) = 1.0
C--- CMSFL(1) IS CONVERSION FACTOR FROM (KG/HR) TO (LBM/HR)
      CMSFL(1) = 2.67924
      CMSFL(2) = 1.0
C--- CTMPF(1) IS CONVERSION FACTOR FROM (K) TO (R)
      CTMPF(1) = 1.8
      CTMPF(2) = 1.0
C--- CTCON(1) IS CONVERSION FACTOR FROM (WATTS/M K) TO (BTU/HR FT R)
      CTCON(1) = .57817
      CTCON(2) = 1.0
C--- CDEN(1) IS CONVERSION FACTOR FROM (KG/M**3) TO (LBM/FT**3)
      CDEN(1) = .06243
```

```
      CDEN(2) = 1.0
C---- CSPHT(1) IS CONVERSION FACTOR FROM (J/KG K) TO (BTU/LBM R)
      CSPHT(1) = .000239
      CSPHT(2) = 1.0
C---- CVISC(1) IS CONVERSION FACTOR FROM (PA SEC) TO (LBM/FT HR)
      CVISC(1) = 2419.096
      CVISC(2) = 1.0
C---- CGASC(1) IS CONVERSION FROM (J/KG K) TO (FT LBF/LBM R)
      CGASC(1) = .18602
      CGASC(2) = 1.0
C---- CRHOVG IS CONVERSION FROM (KG/SEC M**2) TO (LBM/SEC FT**2)
      CRHOVG(1) = .0204823
      CRHOVG(2) = 1.0
C
      DO 105 I = 1,30
105   TITLE(I) = TIKLE
C
      DO 106 I = 1,1000
      BCHGS(I) = 0.0
      BCHGP(I) = 0.0
      BCTGS(I) = 0.0
      BCTGP(I) = 0.0
      BCQGS(I) = 0.0
      BCQGP(I) = 0.0
      BCPGS(I) = 0.0
106   BCPGP(I) = 0.0
      RHOC = 0.0
      RHOM = 0.0
      SPHTC = 0.0
      SPHTM = 0.0
      DO 107 I = 1,400
      THUBIN(I) = 0.0
      QHUBIN(I) = 0.0
      TTIPIN(I) = 0.0
      QTIPIN(I) = 0.0
      RHOVG(I) = 0.0
107   PEX(I) = 0.0
      DO 108 I = 1,50
      BCTIME(I) = 0.0
      TTIO(I) = 0.0
      PTIO(I) = 0.0
      WSVST(I) = 0.0
108   RR(I) = 0.0
C
      DO 110 I = 1,6000
110   CHANL(I) = 0.0
C
      DO 112 I = 1,15
      PEXIT(I) = 0.0
      DO 112 J = 1,80
      AKC(I,J) = 0.0
112   AKW(I,J) = 0.0
C
      DO 115 I = 1,2000
115   INDCHN(I) = 0
      DO 116 I = 1,20
      AKCTBL(I) = 0.0
116   AKWTBL(I) = 0.0
      IPLOT = 0
      IWRITE = 0
```

```
      INEDIT = 0
      TEPS = 1.0
      DLTYME = 0.0
      READ(5,TITL)
      READ(5,CHANLS)
      READ(5,BC)
      IF (BCHGS(1).EQ.0.0) BTA=1.0
      IF (BCQGS(1) .EQ.0.0) BTA = 0.0
      IF (TTIPIN(1).GT.0.0) ITIP = 1
      IF (THUBIN(1).GT.0.0) IHUB = 1
      IF (ABS(QTIPIN(1)).GT.0.0) ITIP = 3
      IF (ABS(QHUBIN(1)).GT.0.0) IHUB = 3
      WS = WSVST(1)
C
      PEXIT(1) = PEX(1)
      DO 175 ICHLNO = 1,NSLICE
C
C  ICHLNO IS THE CHANNEL NUMBER;  = 1 AT THE HUB,   = NSLICE AT THE TIP
C
      READ(5,CONTRL)
      NODSF = 5*NFWD
C--NODSF IS THE NUMBER OF NODES IN THE FORWARD REGION
C
      NODST = 5*NSTA
C--NODST IS THE TOTAL NUMBER OF NODES IN THE BLADE SLICE ICHLNO
C
      NBLKSZ = (15 + 2*NODST) + 8*NSTA
C--NBLKSZ IS THE SIZE OF THE DATA BLOCK RESERVED IN CHANL ARRAY FOR THIS
C  SLICE ICHLNO
C
      ISBLOK = IEND + 1
C--ISBLOK IS THE STARTING POINT IN CHANL ARRAY FOR THIS BLOCK OF DATA
C
      INSTRT = 15 + (ICHLNO-1)*(15 + NSTA)
C--INSTRT IS THE STARTING POINT IN INDCHN ARRAY FOR THIS BLOCK OF
C  INTEGER DATA
C
      INDCHN(ICHLNO) = INSTRT
      INDCHN(INSTRT) = ICHLNO
      INDCHN(INSTRT+1) = IFILM
      INDCHN(INSTRT+2) = ICOR
      INDCHN(INSTRT+3) = NFWD
      INDCHN(INSTRT+4) = NSTA
      INDCHN(INSTRT+5) = ISBLOK
      INDCHN(INSTRT+6) = NBLKSZ
      INDCHN(INSTRT+7) = IPLOT
      INDCHN(INSTRT+8) = MD1
      INDCHN(INSTRT+9) = MD2
      INDCHN(INSTRT+10) = MD3
      INDCHN(INSTRT+12) = IHUB
      INDCHN(INSTRT+13) = ITIP
      IIHCTZ = INSTRT + 14
C--IIHCTZ IS THE RELATIVE ZERO POINT IN INDCHN FOR STORAGE OF THE
C  INDICATOR IHC
      READ(5,PROPS)
      S(ICHLNO) = SPAN*CINCH(IUNITS)
      APLN(ICHLNO) = APLEN*CINCH(IUNITS)*CINCH(IUNITS)
      DPLN(ICHLNO) = DHYD*CINCH(IUNITS)
      ROUT(ICHLNO) = RO*CINCH(IUNITS)
      RIN(ICHLNO) = RI*CINCH(IUNITS)
```

```
C
C  NOW, /GEO/ IS READ, NGEO TIMES, AND THE DATA STORED IN CHANL ARRAY.
C
C
C  ISBLOK = THE STARTING POINT IN CHANL ARRAY FOR CHANNEL ICHLNO DATA
C
C  FIRST, STORE THE SINGLE VALUED DATA
C
      CHANL(ISBLOK) = CD
      CHANL(ISBLOK+1)  = ALPHA
      CHANL(ISBLOK+2)  = BETA
      CHANL(ISBLOK+3)  = DELTA
      CHANL(ISBLOK+4)  = EPS
      CHANL(ISBLOK+6)  = ADUMP*CINCH(IUNITS)**2
      CHANL(ISBLOK+7)  = SPAN*CINCH(IUNITS)
      CHANL(ISBLOK+8)  = BTA
      CHANL(ISBLOK+9)  = DLTYME
      CHANL(ISBLOK+10)= TEPS
C
C THEN THE ARRAYS ARE STORED:
C
C THE FOLLOWING ARE STORED BY NODE NUMBER:
C     THK  (TAU),    TDLX  (DLX)
C THE REST ARE STORED BY STATION NUMBER:
C     TDHJ  (DHJ),     TDHF  (DHF),     TXN  (XN),     TRR  (RR),
C         TDP  (DP),     TSP  (SP),
C       (AKC),(AKW),     IHCT  (IHC).
C
      ITHKZ  = ISBLOK + 14
      ITDLXZ = ISBLOK + 14 + NODST
      ITDHJZ = ISBLOK + 14 + 2*NODST
      ITDHFZ = ISBLOK + 14 + 2*NODST + NSTA
      ITXNZ  = ISBLOK + 14 + 2*NODST + 2*NSTA
      ITRRZ  = ISBLOK + 14 + 2*NODST + 3*NSTA
      ITDPZ  = ISBLOK + 14 + 2*NODST + 4*NSTA
      ITSPZ  = ISBLOK + 14 + 2*NODST + 5*NSTA
      IEND   = ISBLOK + 14 + 2*NODST + 8*NSTA
      THK(1) = 0.0
      DO 170 I = 1,NGEO
      ISTB = 0
      READ(5,GEO)
      IF (THK(1).LE.0.0) THK(1) = .0001*THK(2)
      IF (TDLX(1).GT.2.0*TDLX(4).OR.TDLX(4).GT.1.2*TDLX(1))
     Z                                            WRITE(8,136) ICHLNO,ISTA
136   FORMAT(/'  CHANNEL ',I2,',   STATION ',I3,
     Z                        ', ---TDLX VALUES DO NOT LOOK RIGHT')
      IF (TDHJ.GT.0..AND.TXN.LT.1.1*TDHJ) WRITE(8,137) ICHLNO,ISTA
137   FORMAT(/'  CHANNEL ',I2,',   STATION ',I3,
     Z                ', ---HOLE SPACING AND DIAMETER DO NOT LOOK RIGHT')
      IF (ISTB.EQ.0) ISTB = ISTA
      DO 165 J = ISTA,ISTB,2
C
C   J REPRESENTS THE STATION NUMBER IN THIS CASE
C
      JSENS = J - 2*(J/2)
C
C   JSENS = 0 INDICATES THAT STATION NO. IS EVEN AND STATION IS ON
C         SUCTION SIDE
C   JSENS = 1 INDICATES THAT STATION NO. IS ODD AND STATION IS ON
C         PRESSURE SIDE
```

```
C
      IARG = ITDHJZ + J
      CHANL(IARG) = TDHJ*CINCH(IUNITS)
      IARG = ITDHFZ + J
      CHANL(IARG) = TDHF*CINCH(IUNITS)
      IARG = ITXNZ + J
      CHANL(IARG) = TXN*CINCH(IUNITS)
      IARG = ITRRZ + J
      CHANL(IARG) = TRR*CINCH(IUNITS)
      IARG = ITDPZ + J
      CHANL(IARG) = TDP*CINCH(IUNITS)
      IARG = ITSPZ + J
      CHANL(IARG) = TSP*CINCH(IUNITS)
      IARG = IIHCTZ + J
      INDCHN(IARG) = IHCT
      NODOUT = 5*J - 4
C
C    NODOUT IS THE NODE NO. ON THE OUTSIDE SURFACE AT STATION J
C    5 IS THE NUMBER OF NODES AT STATION J
C
145   CONTINUE
      LOCA = ITDLXZ + NODOUT
      IF (TDLX(3).LE.0.) GO TO 155
      DO 150 L = 1,5
      LOCAL = LOCA + L - 1
150   CHANL(LOCAL) = TDLX(L)*CINCH(IUNITS)
      GO TO 160
155   CHANL(LOCA) = TDLX(1)*CINCH(IUNITS)
      CHANL(LOCA+3) = TDLX(4)*CINCH(IUNITS)
      AA = TDLX(1)
      B = (TDLX(4)-TDLX(1))/(THK(1)+THK(2))
      CHANL(LOCA+1) = (AA + B*THK(1))*CINCH(IUNITS)
      CHANL(LOCA+2) = (AA + B*(THK(1)+THK(2)/2.))*CINCH(IUNITS)
      CHANL(LOCA+4) = (AA + B*(THK(1)+THK(2)+THK(3)/2.))*CINCH(IUNITS)
160   CONTINUE
      LOCA = ITHKZ + NODOUT
      CHANL(LOCA) = THK(1)*CINCH(IUNITS)
      CHANL(LOCA+2) = THK(2)*CINCH(IUNITS)
      CHANL(LOCA+4) = THK(3)*CINCH(IUNITS)
165   CONTINUE
170   CONTINUE
175   CONTINUE
C
C---  CONVERT UNITS ON BC DATA
C
      IF (IUNITS.EQ.2) GO TO 300
      NTBC = 1
      DO 205 I = 2,50
      IF (BCTIME(I).LE.0.0) GO TO 210
205   NTBC = NTBC + 1
210   NSETS = NBCS*NSLICE*NTBC
      NSETP = NBCP*NSLICE*NTBC
C
      DO 215 I = 1,NSETS
      BCXS(I) = BCXS(I)*CINCH(1)
      BCHGS(I) = BCHGS(I)*CHTC(1)
      BCTGS(I) = BCTGS(I)*CTMPF(1) - 460.
      BCQGS(I) = BCQGS(I)*CHFLX(1)
215   BCPGS(I) = BCPGS(I)*CPRSR(1)
C
```

```
      DO 220 I = 1,NSETP
      BCXP(I) = BCXP(I)*CINCH(1)
      BCHGP(I) = BCHGP(I)*CHTC(1)
      BCTGP(I) = BCTGP(I)*CTMPF(1) - 460.
      BCQGP(I) = BCQGP(I)*CHFLX(1)
220   BCPGP(I) = BCPGP(I)*CPRSR(1)
C
      NSET = NSTA*NTBC
      DO 225 I = 1,NSET
      RHOVG(I) = RHOVG(I)*CRHOVG(1)
      THUBIN(I) = THUBIN(I)*CTMPF(1) - 460.
      QHUBIN(I) = QHUBIN(I)*CHFLX(1)
      TTIPIN(I) = TTIPIN(I)*CTMPF(1) - 460.
225   QTIPIN(I) = QTIPIN(I)*CHFLX(1)
C
      NSET = NSLICE*NTBC
      DO 230 I = 1,NSET
230   PEX(I) = PEX(I)*CPRSR(IUNITS)
C
      DO 235 I = 1,49,2
      TTIO(I) = TTIO(I)*CTMPF(IUNITS) - 460.
235   PTIO(I) = PTIO(I)*CPRSR(IUNITS)
      WPLEN = WPLEN*CMSFL(IUNITS)
      RHOC = RHOC*CDEN(IUNITS)
      RHOM = RHOM*CDEN(IUNITS)
      SPHTC = SPHTC*CSPHT(IUNITS)
      SPHTM = SPHTM*CSPHT(IUNITS)
C
      DO 280 I = 1,19,2
      AKCTBL(I) = AKCTBL(I)*CTMPF(IUNITS) - 460.
      AKCTBL(I+1) = AKCTBL(I+1)*CTCON(IUNITS)
      AKWTBL(I) = AKWTBL(I)*CTMPF(IUNITS) - 460.
280   AKWTBL(I+1) = AKWTBL(I+1)*CTCON(IUNITS)
C
300   CONTINUE
C
C
      IF (IFILM.NE.2) GO TO 320
      DO 310 I = 1,NSTA
310   RHOVGA(I) = RHOVG(I)
C
320   CONTINUE
C
C  IF INEDIT .GT. 0,  PRINT AN INPUT EDIT
C
      DO 180 I = 1,NSLICE
180   CALL INPRT(I,INEDIT)
185   CONTINUE
      RETURN
      END


C----SOURCE.NHCFRCT
      FUNCTION HCFRCD(IS,LCOOL,LIN)
C
C-   SOURCE.NHCFRCT----
C
      COMMON /TCO/ ADUMP,     BTA,       CD,        CP,
     Z             GAM,       PIM,       R,         SPAN,       TOG,
     Z             WDUMP,     WIM,       AKC(15,80), AKW(15,80),
     Z             A(400),    AJET(80),  AM2(80),   CNUM(80),
     Z             DH(80),    DHF(80),   DHJ(80),
```

```
     Z                DLX(400),  FF(80),     HC(80),     HG(80),
     Z                P(2,15,80),PEXIT(15),  PUMP(80),   QG(80),
     Z                QSNK(80),  RR(80),     S(15),      T(2,15,400),
     Z                TG(80),    TAU(400),   WFC(80),
     Z                WJ(15,80), WCROS(2,15,80),         XN(80),
     Z                ICOR,      IFILM,      IHUB,       ITIP,
     Z                ISBLOK,    ISLICE,     NBLKSZ,     NSLICE,
     Z                NFWD,      NSTA,       IHC(80)
C
C  COMPUTE TURBULENT HEAT TRANSFER COEFFICIENT IN CHANNEL FLOW:
C     NU = .023*( RE**.8 )*( PD**.333 )
C
100   CONTINUE
      TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
      RE = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU)
      HCFRCD = .023*12.*(C/DH(IS))*(RE**.8)*(PD**.333)
200   CONTINUE
      RETURN
      END


C----SOURCE.NHCOOLT
      SUBROUTINE HCOOL(JS)
C
C-    SOURCE.NHCOOLT----
C
C
      COMMON /IMPCOR/ CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, CIMP6, CIMP7,
     Z                DIMP1, DIMP2, DIMP3, DIMP4, DIMP5, DIMP6
C
      COMMON /PRPS/ CPO,       GAMO,       DP(80),     SP(80),  RE(80),
     Z              CPC(80),   GAMC(80),   DUMR1(80),  DUMR2(80)
C
      COMMON /TCO/ ADUMP,      BTA,        CD,         CP,
     Z                GAM,       PIM,        R,          SPAN,       TOG,
     Z                WDUMP,     WIM,        AKC(15,80), AKW(15,80),
     Z                A(400),    AJET(80),   AM2(80),    CNUM(80),
     Z                DH(80),    DHF(80),    DHJ(80),
     Z                DLX(400),  FF(80),     HC(80),     HG(80),
     Z                P(2,15,80),PEXIT(15),  PUMP(80),   QG(80),
     Z                QSNK(80),  RR(80),     S(15),      T(2,15,400),
     Z                TG(80),    TAU(400),   WFC(80),
     Z                WJ(15,80), WCROS(2,15,80),         XN(80),
     Z                ICOR,      IFILM,      IHUB,       ITIP,
     Z                ISBLOK,    ISLICE,     NBLKSZ,     NSLICE,
     Z                NFWD,      NSTA,       IHC(80)
C
      DIMENSION  IGG(80), IRE(80), REJ(80), REJOVR(80)
1     CONTINUE
      TMP=TOG
      CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)
      CONDCT = C
      XMUTOG = XMU
      PDTOG  = PD
      PI=3.14159
      IF (JS.GT.1) GO TO 101
      IF (ICOR.EQ.1) GO TO 101
      IF (WJ(ISLICE,JS).LE.0.0) GO TO 101
C
C--- LEADING EDGE HEAT TRANSFER CORRELATION FOR STATIONS FORWARD OF ICOR
C--- CORRELATION OF  METZGER ET AL, J. ENG. POWER, JULY 1969,PP 149-158
```

```
C
5     NEND=ICOR+1
      XS = 0.0
      XP = 0.0
      DO 50 J = 3,NEND,2
      LINS = 5*(J-1) - 1
      LINP = 5*J - 1
      XS = XS + DLX(LINS)
      XP = XP + DLX(LINP)
50    CONTINUE
55    XL = (XS + XP)/2.
      IF (AJET(JS).GT.0.) GMASS = WJ(ISLICE,JS)/AJET(JS)
60    BES = PI*DHJ(JS)**2/(4.*XN(JS))
      DEH = 2.*BES
65    REJ(JS) = 12.*3600.*GMASS*DEH/XMU
      PROD = REJ(JS)**.27*(XL/BES)**.52
70    STANMX = .355/PROD
      HC(JS) = STANMX*CP*GMASS*144.*3600.
C
      IF (REJ(JS).LT.1150..OR.REJ(JS).GT.6300.) WRITE(6,75) REJ(JS)
   75 FORMAT(1H /'***WARNING*** LEADING EDGE IMPINGEMENT JET REYNOLDS ',
     Z      'NUMBER IS ',F8.1/'    RANGE OF THE CORRELATION IS 1150',-
     Z      ' < REJ < 6300')
      ILEAD = ICOR - 1
      IF (ILEAD.LT.2) GO TO 85
      DO 80 I = 2,ILEAD
      IF(WJ(ISLICE,I).GT.0.0) GO TO 90
80    HC(I) = HC(JS)
85    CONTINUE
      GO TO 101
90    WRITE(8,95) ICOR
95    FORMAT(//'    SOLUTION TERMINATED***TOO MANY ROWS OF IMPINGEMENT',
     Z         ' HOLES FORWARD OF STATION',I3,'. HOLES ARE ',
     Z    'ALLOWED ONLY AT STATION 1.')
      STOP
C
C--KIRCHER-TABAKOFF CORRELATION, IMPINGEMENT WITH CROSS FLOW
C--ICOR = STATION NUMBER APPLICATION OF THIS CORRELATION BEGINS
C
101   IGGC = 0
      IREC = 0
      ISTRT=ICOR
      IF (JS.GT.1) ISTRT= 1
C
      IF (CIMP1.NE.0.0) GO TO 400
      DO 130 I = ISTRT,NFWD
      WC = ABS(WCROS(2,ISLICE,I))
      II = 5*I
      REJ(I) = 0.0
      IF (IHC(I).EQ.1) GO TO 103
      LCOOL = 5*I
      LIN   = LCOOL - 1
      HC(I) = HCFRCD(I,LCOOL,LIN)
      GO TO 130
103   CONTINUE
      IF (AJET(I).EQ.0.0) GO TO 128
      IF (WJ(ISLICE,I).LE.0.) GO TO 128
      TMP=(T(2,ISLICE,LIN)+TOG)/2.
      CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)
      CONDCT = C
```

```
      XMUTOG = XMU
      PDTOG  = PD
105   REJ(I)=WJ(ISLICE,I)/AJET(I)*DHJ(I)/(XMUTOG/3600.)*12.0
      GG=(WC/A(II))/(WJ(ISLICE,I)/AJET(I))
      IF (GG.LE.2.0) GO TO 110
      IGGC = IGGC + 1
      IGG(IGGC) = I
110   CONTINUE
      IF (REJ(I).GE.300.0.AND.REJ(I).LE.3.E4) GO TO 115
      IREC = IREC + 1
      IRE(IREC) = I
115   CONTINUE
      IF (REJ(I).LT.3000.) GO TO 120
      AM=-.002517*(XN(I)/DHJ(I))**2+.068485*XN(I)/DHJ(I)+.506994
      HC(I)=REJ(I)**AM
      HC(I)=HC(I)*EXP(.02596*(XN(I)/DHJ(I))**2-.8259*XN(I)/DHJ(I)+.3985)
      HC(I)=HC(I)/(1.+.4696*
     Z       ((WC/A(II))/(WJ(ISLICE,I)/AJET(I))*TAU(II)/DHJ(I))**.965)
      GO TO 125
120   AM=-.001452*(XN(I)/DHJ(I))**2+.042838*(XN(I)/DHJ(I))+.516548
      HC(I)=REJ(I)**AM
      HC(I)=HC(I)*EXP(.0126*(XN(I)/DHJ(I))**2-.5106*XN(I)/DHJ(I)-.2057)
      HC(I)=HC(I)/(1.+.4215*
     Z       ((WC/A(II))/(WJ(ISLICE,I)/AJET(I))*TAU(II)/DHJ(I))**.58)
125   CONTINUE
      HC(I)=HC(I)*CONDCT/DHJ(I)*12.0*PDTOG**.33*(TAU(II)/DHJ(I))**.091
      GO TO 130
C
128   CONTINUE
      IF (I.GT.2) HC(I) =  HC(I-2)
      IF (I.EQ.2) HC(I) =  HC(1)
      IF (I.EQ.1) HC(I) =  HC(3)
130   CONTINUE
      IST = NFWD + 1
      DO 150 I = IST,NSTA,2
      IF (IHC(I).NE.1) GO TO 155
150   HC(I) = HC(I-2)
155   IST = NFWD+2
      DO 160 I = IST,NSTA,2
      IF (IHC(I).NE.1) GO TO 165
160   HC(I) = HC(I-2)
165   CONTINUE
      IF (IGGC.GT.0) WRITE(6,140) (IGG(I),I=1,IGGC)
      DO 132 I = 1,IREC
      ISTATN = IRE(I)
      REJOVR(I) = REJ(ISTATN)
132   CONTINUE
      IF (IREC.GT.0) WRITE(6,145) (IRE(I),REJOVR(I),I=1,IREC)
135   CONTINUE
140   FORMAT(1H /' ******* WARNING ******* RATIO OF CROSSFLOW TO ',
     Z       'JET-FLOW IS OUT OF THE RANGE OF ',
     Z     'THE CORRELATION AT THE FOLLOWING STATIONS: '/23X,20(I4,','))
145   FORMAT(1H /' ******* WARNING ******* JET REYNOLD''S NUMBER IS ',
     Z       'OUT OF THE RANGE OF THE CORRELATION ',
     Z     'AT THE FOLLOWING STATIONS: '/1X,8('**',I2,'--',F8.1,'*'))
      DO 301 I = 1,NFWD
      DUMR2(I) = REJ(I)
301   CONTINUE
      RETURN
C
```

```
C--- GENERAL CORRELATION FOR IMPINGEMENT WITH CROSSFLOW IS EVALUATED HER
C    E
C--- FORM OF CORRELATION IS:
C---ST = CIMP1*(GG**CIMP2)*(GI**CIMP3)*((Z/D)**CIMP4)
C       *((X/D)**CIMP5)*(REJ**CIMP6)*(PDTOG**CIMP7)
C---     WHERE GG IS THE MASS FLUX RATIO, FREE STREAM TO JET, AND
C---             GI IS THE MOMENTUM FLUX RATIO.
C
400   CONTINUE
      DO 450 I = ISTRT,NFWD
      WC = ABS(WCROS(2,ISLICE,I))
      II = 5*I
      ROINVJ = R*TOG/(144.*P(2,ISLICE,I))
      ROINVC = R*T(2,ISLICE,II)/(144.*P(2,ISLICE,I))
      REJ(I) = 0.0
      IF (IHC(I).EQ.1) GO TO 403
      LCOOL = 5*I
      LIN   = LCOOL - 1
      HC(I) = HCFRCD(I,LCOOL,LIN)
      GO TO 450
403   CONTINUE
      IF (AJET(I).EQ.0.0) GO TO 445
      IF (WJ(ISLICE,I).LE.0.) GO TO 445
      TMP=(T(2,ISLICE,LIN)+TOG)/2.
      CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)
      CONDCT = C
      XMUTOG = XMU
      PDTOG  = PD
405   REJ(I)=WJ(ISLICE,I)/AJET(I)*DHJ(I)/(XMUTOG/3600.)*12.0
      GG=(WC/A(II))/(WJ(ISLICE,I)/AJET(I))
      GI = ((WC/A(II))**2*ROINVC)/((WJ(ISLICE,I)/AJET(I))**2*ROINVJ)
      ZOVERD = TAU(II)/DHJ(I)
      XOVERD = XN(I)/DHJ(I)
      ST = CIMP1*(GG**CIMP2)*(GI**CIMP3)*(ZOVERD**CIMP4)
     Z      *(XOVERD**CIMP5)*(REJ(I)**CIMP6)*(PDTOG**CIMP7)
      HC(I) = 144.*3600.*ST*CP*WJ(ISLICE,I)/AJET(I)
      GO TO 450
C
445   CONTINUE
      IF (I.GT.2) HC(I) =  HC(I-2)
      IF (I.EQ.2) HC(I) =  HC(1)
      IF (I.EQ.1) HC(I) =  HC(3)
450   CONTINUE
      IST = NFWD+1
      DO 460 I = IST,NSTA,2
      IF (IHC(I).NE.1) GO TO 465
460   HC(I) = HC(I-2)
465   IST = NFWD+2
      DO 470 I = IST,NSTA,2
      IF (IHC(I).NE.1) GO TO 475
470   HC(I) = HC(I-2)
475   CONTINUE
C
      DO 485 I = 1,NFWD
      DUMR2(I) = REJ(I)
485   CONTINUE
      RETURN
      END


C----SOURCE.NHCPINT
      SUBROUTINE HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EFAREA)
```

```
C
C-    SOURCE.NHCPINT----
C
      COMMON /PRPS/ CPO,       GAMO,       DP(80),     SP(80),  RE(80),
     Z              CPC(80),   GAMC(80),   DUMR1(80),  DUMR2(80)
C
      COMMON /TCO/ ADUMP,      BTA,        CD,         CP,
     Z             GAM,        PIM,        R,          SPAN,       TOG,
     Z             WDUMP,      WIM,        AKC(15,80), AKW(15,80),
     Z             A(400),     AJET(80),   AM2(80),    CNUM(80),
     Z             DH(80),     DHF(80),    DHJ(80),
     Z             DLX(400),   FF(80),     HC(80),     HG(80),
     Z             P(2,15,80),PEXIT(15),   PUMP(80),   QG(80),
     Z             QSNK(80),   RR(80),     S(15),      T(2,15,400),
     Z             TG(80),     TAU(400),   WFC(80),
     Z             WJ(15,80),  WCROS(2,15,80),         XN(80),
     Z             ICOR,       IFILM,      IHUB,       ITIP,
     Z             ISBLOK,     ISLICE,     NBLKSZ,     NSLICE,
     Z             NFWD,       NSTA,       IHC(80)
C
      DIMENSION EFAREA(80), DELTAN(15)
C
C   COMPUTE THE HEAT TRANSFER COEFFICIENT AND EFFECTIVENESS FOR A
C   TRIANGULAR ARRAY OF PIN FINS
C
C   WHERE DP IS PIN DIAMETER IN INCHES AND SP IS PIN SPACING IN INCHES
C
      VDP = DP(IS)
      VSP = SP(IS)
100   CONTINUE
      TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
C-- NO. OF PINS AT THIS STATION IS:
      PINS = SPAN*DLX(LCOOL)/(.86603*VSP**2)
C-- AVERAGE LENGTH OF PINS:
      SLP = (TAU(LCOOL) + TAU(LCUP))/2.
      IF (IS.GT.NFWD.AND.IS.LE.NFWD+2) SLP = TAU(LCOOL)
C-- MINIMUM FLOW AREA:
      AMIN = SLP*SPAN*(VSP-VDP)/VSP
C-- TOTAL SURFACE AREA:
      AHTTR = 2.*DLX(LCOOL)*SPAN + 3.14159*PINS*(VDP*SLP-VDP**2/4.)
C-- CHANNEL HYDRAULIC DIAMETER:
      DH(IS) = 4.*AMIN*DLX(LCOOL)/AHTTR
      REDH = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(AMIN*XMU)
      TERM1 = -.89*(VSP/SLP)**.5075
      TERM2 = -3.094*VDP/VSP
      TERM3 = 4.143*EXP(TERM1 + TERM2)/(REDH**.2946)
105   CONTINUE
      HC(IS) = (12.*C/DH(IS))*(.023 + TERM3)*(REDH**.8)*(PD**.333)
      EML = SQRT(4.*HC(IS)*SLP**2/(AKW(ISLICE,IS)*VDP))
      EFTVNS = TANH(EML)/EML
C-- CHECK LOCATION OF HEAT FLOW SPLIT POINT IF THIS IS A TRAILING
C   EDGE REGION STATION
C
      IF (IS.LE.NFWD) GO TO 160
      TBAR = (T(2,ISLICE,LCOOLP)-T(2,ISLICE,LCOOL))/
     Z              (T(2,ISLICE,LIN)-T(2,ISLICE,LCOOL))
      HYCOS = COSH(EML)
      HYSIN = SINH(EML)
      IF (HYCOS-TBAR.LT.HYSIN) GO TO 120
```

```
      WRITE(6,110) LCOOLP,LIN,DELTAN(ISLICE)
  110 FORMAT(1H //' **** WARNING **** NODE',I3,
     Z        ' IS RECEIVING HEAT FROM NODE',I3,' THROUGH THE PINS.',
     Z           ' RESULTS ARE INVALID. DELTAN =',F7.4)
      GO TO 140
  120 CONTINUE
      IF (HYCOS-TBAR.GT.0.) GO TO 130
      WRITE(6,110) LIN,LCOOLP,DELTAN(ISLICE)
      GO TO 140
  130 CONTINUE
      XOVRL = (HYCOS-TBAR)/HYSIN
      XOVRL = ALOG((1.+XOVRL)/(1.-XOVRL))/(2.*EML)
  140 CONTINUE
  160 CONTINUE
      EFAREA(IS) = DLX(LIN)*SPAN
     Z               - 3.14159*PINS*(VDP**2/4.- EFTVNS*VDP*SLP*XOVRL)
      IF (IS.GT.NFWD) EFAREA(IS+1) = DLX(LCOOLP)*SPAN
     Z             - 3.14159*PINS*(VDP**2/4.- EFTVNS*VDP*SLP*(1.-XOVRL))
      IF (IS.LE.NFWD)  EFAREA(IS) = DLX(LIN)*SPAN
     Z                      - 3.14159*PINS*(VDP**2/4.- EFTVNS*VDP*SLP)
  170 CONTINUE
      RETURN
      END


C----SOURCE.NINPRTT
      SUBROUTINE INPRT(ICHNL,INEDIT)
C
C-    SOURCE.NINPRTT----
C
      COMMON /BOUND/ BCXS(400),   BCXP(400),   BCHGS(1000), BCHGP(1000),
     Z               BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),
     Z               BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),
     Z               QHUBIN(400), QHUB(80),    TTIPIN(400), TTIP(80),
     Z               QTIPIN(400), QTIP(80),    RHOVG(400),  PEX(400),
     Z               BCTIME(50),  TTIO(50),    PTIO(50),    WPLEN,
     Z               WSVST(50),   AKCTBL(20),  AKWTBL(20),  NBCS,  NBCP
C
      COMMON /GAAS/ GS(200),  NG
C
      COMMON /PRPS/ CPO,      GAMO,     DP(80),   SP(80),  RE(80),
     Z              CPC(80),  GAMC(80), DUMR1(80), DUMR2(80)
C
      COMMON /RADL/ APLN(15), DPLN(15),  RIN(15),   ROUT(15),
     Z              PIN(15),  TIN(15),   W(15),     WS
      COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),
     Z                IPLOT,  MD1,  MD2,  MD3, IADJIN, IWRITE
C
      COMMON /TCO/ ADUMP,      BTA,       CD,          CP,
     Z             GAM,        PIM,       R,           SPAN,       TOG,
     Z             WDUMP,      WIM,       AKC(15,80),  AKW(15,80),
     Z             A(400),     AJET(80),  AM2(80),     CNUM(80),
     Z             DH(80),     DHF(80),   DHJ(80),
     Z             DLX(400),   FF(80),    HC(80),      HG(80),
     Z             P(2,15,80),PEXIT(15),  PUMP(80),    QG(80),
     Z             QSNK(80),   RR(80),    S(15),       T(2,15,400),
     Z             TG(80),     TAU(400),  WFC(80),
     Z             WJ(15,80),  WCROS(2,15,80),         XN(80),
     Z             ICOR,       IFILM,     IHUB,        ITIP,
     Z             ISBLOK,     ISLICE,    NBLKSZ,      NSLICE,
     Z             NFWD,       NSTA,      IHC(80)
C
```

```
      COMMON /TRNSNT/ RHOC,        RHOM,        SPHTC,        SPHTM,
     Z                DLTYME,      TYME,        TEPS,         TYMMAX
C
      COMMON /UNITS/ CINCH(2), CHTC(2),  CHFLX(2), CPRSR(2), CMSFL(2),
     Z               CTMPF(2), CTCON(2), CDEN(2),  CSPHT(2), CGASC(2),
     Z               CVISC(2), CRHOVG(2),IUNITS
C
      DIMENSION DUM1(10),DUM2(10),DUM3(10),DUM4(10),DUM5(10),DUM6(10),
     Z          DUM7(10),DUM8(10),DUM9(10),DUM53(10),DUM55(10)
      DIMENSION DUM10(10),DUM11(10),DUM12(10),DUM13(10),DUM14(10),
     Z          DUM15(10),DUM16(10),DUM25(10),DUM52(10)
      DIMENSION DUM17(10),DUM18(10),DUM19(10),DUM20(10)
      DIMENSION NFLUID(200), HCAL(4), UL(2), UA(2)
      DATA HCAL/'IMPG','CHAN','PINS','    '/
      DATA UL/' CM ',' IN '/
      DATA UA/' CM*',' IN*'/
      CALL PREP(ICHNL,NZON,1)
C
C  INITIALIZE TEMPERATURE DISTRIBUTION  (DEGREES R)
C
      I = ICHNL
      NODSF = 5*NFWD
      NODSTM = 5*NSTA - 4
      NODST = 5*NSTA
      DO 830 I1 = 5,NODSF,5
      IS = I1/5
      LO = I1-4
      LJ = I1-3
      L  = I1-2
      LI = I1-1
      T(2,I,LO) = .9*TG(IS)
      T(2,I,LI) = T(2,I,LO)/1.08
      T(2,I,LJ) = T(2,I,LO) - (T(2,I,LO)-T(2,I,LI))*TAU(LO)/(TAU(LO)+
     Z                                                          TAU(L))
      T(2,I,L)  = T(2,I,LO) - (T(2,I,LO)-T(2,I,LI))*(TAU(LO)+TAU(L)/2.)/
     Z                                                  (TAU(LO)+TAU(L))
830   T(2,I,I1) = TTIO(1) + 460.
      ISTRT = NODSF + 5
      DO 860 I1 = ISTRT,NODSTM,10
      T(2,I,I1) = T(2,I,NODSF)
      T(2,I,I1+5) = T(2,I,I1)
      DO 860 J = 1,4
      IPJ  = I1 + J
      IMJ  = I1 + J - 5
      IUPP = NODSF + J - 5
      T(2,I,IPJ) = T(2,I,IUPP)
860   T(2,I,IMJ) = T(2,I,IUPP)
      DO 865 J = 1,NODST
865   T(1,I,J) = T(2,I,J)
C
      IF (ICHNL.GT.1) GO TO 94
      WRITE(6,408)
408   FORMAT(1H1,////,20X,'PROPERTY TABLES'///)
      WRITE(6,410)
410   FORMAT(1H ,'OUTER COATING EFFECTIVE THERMAL CONDUCTIVITY')
C
      IF (IUNITS.EQ.1) GO TO 420
C
      WRITE(6,412) (AKCTBL(I),I=1,19,2)
      WRITE(6,414) (AKCTBL(I),I=2,20,2)
```

```
412    FORMAT(//5X,'T, (F)',10X,10(F9.1))
414    FORMAT(5X,'K, (BTU/HR/FT/R)',10(F9.3))
       WRITE(6,416)
416    FORMAT(////' WALL METAL THERMAL CONDUCTIVITY')
       WRITE(6,412) (AKWTBL(I),I=1,19,2)
       WRITE(6,414) (AKWTBL(I),I=2,20,2)
C
       GO TO 445
C
420    CONTINUE
       DO 418 I = 1,19,2
       AKCTBL(I) = (AKCTBL(I)+460.)/1.8
       AKWTBL(I) = (AKWTBL(I)+460.)/1.8
       AKCTBL(I+1) = AKCTBL(I+1)/CTCON(1)
418    AKWTBL(I+1) = AKWTBL(I+1)/CTCON(1)
C
       WRITE(6,422) (AKCTBL(I),I=1,19,2)
       WRITE(6,424) (AKCTBL(I),I=2,20,2)
422    FORMAT(//5X,'T, (K)',4X,10(F9.1))
424    FORMAT(5X,'K, (W/M/K)',10(F9.3))
       WRITE(6,416)
       WRITE(6,422) (AKWTBL(I),I=1,19,2)
       WRITE(6,424) (AKWTBL(I),I=2,20,2)
C
C
       DO 448 I = 1,19,2
       AKCTBL(I) = 1.8*AKCTBL(I) - 460.
       AKWTBL(I) = 1.8*AKWTBL(I) - 460.
       AKCTBL(I+1) = AKCTBL(I+1)*CTCON(1)
448    AKWTBL(I+1) = AKWTBL(I+1)*CTCON(1)
C
445    CONTINUE
C
       WRITE(6,450)
450    FORMAT(1H ////'   TABLE OF GAS PROPERTIES')
       NGS = NG
       IF (NG.GT.10) NGS = 10
C
       IF (IUNITS.EQ.1) GO TO 470
C
       WRITE(6,452) (GS(J),J=1,NGS)
452    FORMAT(//5X,'TEMPERATURE (F)',10(F9.1))
       L = NG + 1
       LE = NG + NGS
       WRITE(6,454) (GS(J),J=L,LE)
454    FORMAT(5X,'K,(BTU/HR/FT/R)',10(F9.5))
       L = 2*NG + 1
       LE = 2*NG + NGS
       WRITE(6,456) (GS(J),J=L,LE)
456    FORMAT(5X,'CP, (BTU/LBM/R)',10(F9.5))
       L = 3*NG + 1
       LE = 3*NG + NGS
       WRITE(6,458) (GS(J),J=L,LE)
458    FORMAT(5X,'PRANDTL  NUMBER',10(F9.5))
       L = 4*NG + 1
       LE = 4*NG + NGS
       WRITE(6,460) (GS(J),J=L,LE)
460    FORMAT(5X,'VIS.(LBM/FT/HR)'10(F9.5))
C
       GO TO 90
```

```
C
C
470   CONTINUE
      DO 471 J = 1,NGS
471   DUM1(J) = (GS(J)+460.)/1.8
      WRITE(6,472) (DUM1(J),J=1,NGS)
472   FORMAT(//5X,'TEMPERATURE (K)',10(F9.1))
      L = NG + 1
      LE = NG + NGS
      JI = 0
      DO 473 J = L,LE
      JI = JI+1
473   DUM1(JI) = GS(J)/CTCON(1)
      WRITE(6,474) (DUM1(J),J=1,JI)
474   FORMAT(5X,'K,        (W/M/K)',10(F9.5))
      L = 2*NG + 1
      LE = 2*NG + NGS
      JI = 0
      DO 475 J = L,LE
      JI = JI+1
475   DUM1(JI) = GS(J)/CSPHT(1)
      WRITE(6,476) (DUM1(J),J=1,JI)
476   FORMAT(5X,'CP,      (J/KG/K)',10(F9.2))
      L = 3*NG + 1
      LE = 3*NG + NGS
      WRITE(6,478) (GS(J),J=L,LE)
478   FORMAT(5X,'PRANDTL  NUMBER',10(F9.5))
      L = 4*NG + 1
      LE = 4*NG + NGS
      JI = 0
      DO 479 J = L,LE
      JI = JI+1
479   DUM1(JI) = GS(J)/CVISC(1)
      WRITE(6,480) (DUM1(J),J=1,JI)
480   FORMAT(5X,'VIS. (N S/M**2)'10(F9.5))
C
90    CONTINUE
      IF (INEDIT.EQ.0) GO TO 350
C
C-- LIST OUT THE INPUT HOT GAS BOUNDARY CONDITIONS--
C
C--MNBC IS THE MAX OF (NBCS & NBCP)
C
      MNBC = NBCS
      IF (MNBC.LT.NBCP) MNBC=NBCP
      NTIMES = 1
C--NTIMES IS THE NUMBER OF TIME STEPS IN BC TABLES
481   IF (BCTIME(NTIMES+1).LE.0.0) GO TO 482
      NTIMES = NTIMES + 1
      GO TO 481
C
482   CONTINUE
      WRITE(6,4820)
4820  FORMAT(1H1,40X,'HOT GAS BOUNDARY CONDITIONS'/)
      WRITE(6,483)
483   FORMAT(' ***************SUCTION SIDE****************',22X,
     Z       '***************PRESSURE SIDE***************')
C
C--SET THE NO. OF POINTS PER TIME STEP IN S&P BC ARRAYS
      NPRTS = NSLICE*NBCS
```

```
      NPRTP = NSLICE*NBCP
C
      NL = 3
      DO 499 IT = 1,NTIMES
C
C-- SET THE NO. OF POINTS THAT PRECEDED TIME STEP 'IT'
      NPRCS = NPRTS*(IT-1)
      NPRCP = NPRTP*(IT-1)
C
C--START THE LOOP THROUGH ALL SLICES
      DO 499 ISL = 1,NSLICE
C
C--SET THE NO. OFF POINTS PRECEDING THIS SLICE
      NBFRS = NPRCS + NBCS*(ISL-1)
      NBFRP = NPRCP + NBCP*(ISL-1)
C
      NL = NL + 3 + MNBC
      IF (NL.LT.60) GO TO 4860
      NL = 3 + MNBC
      WRITE(6,4820)
      WRITE(6,483)
4860  CONTINUE
C
      IF (IT.EQ.1) WRITE(6,484) ISL
484   FORMAT(45X,'INITIAL STEADY STATE'/49X,'SLICE NO.',I2)
      IF (IT.GT.1) WRITE(6,485) BCTIME(IT), ISL
485   FORMAT(45X,'BCTIME =',F8.3,' SEC'/47X,'SLICE NO.',I2)
      WRITE(6,486)
486   FORMAT(4X,'X',7X,'HG',6X,'TG',10X,'QG',6X,'PG',
     Z      26X,'X',7X,'HG',6X,'TG',10X,'QG',6X,'PG')
C
C--HERE WE LOOP WITHIN A SLICE
C
      DO 499 IBC = 1,MNBC
      IF (IBC.GT.NBCS) GO TO 487
      J = NBFRS + IBC
      JXS = (ISL-1)*NBCS + IBC
      TBCXS = BCXS(JXS)/CINCH(IUNITS)
      TBCHGS = BCHGS(J)/CHTC(IUNITS)
      TBCTGS = BCTGS(J)
      IF (IUNITS.EQ.1) TBCTGS=(BCTGS(J)+460.)/1.8
      TBCQGS = BCQGS(J)/CHFLX(IUNITS)
      TBCPGS = BCPGS(J)/CPRSR(IUNITS)
      WRITE(6,489) TBCXS,TBCHGS,TBCTGS,TBCQGS,TBCPGS
489   FORMAT(2X,F6.2,2F8.1,F12.1,F8.1)
487   CONTINUE
      IF (IBC.GT.NBCP) GO TO 499
      J = NBFRP + IBC
      JXP = (ISL-1)*NBCP + IBC
      TBCXP = BCXP(JXP)/CINCH(IUNITS)
      TBCHGP = BCHGP(J)/CHTC(IUNITS)
      TBCTGP = BCTGP(J)
      IF (IUNITS.EQ.1) TBCTGP=(BCTGP(J)+460.)/1.8
      TBCQGP = BCQGP(J)/CHFLX(IUNITS)
      TBCPGP = BCPGP(J)/CPRSR(IUNITS)
      IF (IBC.LE.NBCS) WRITE(6,488) TBCXP,TBCHGP,TBCTGP,TBCQGP,TBCPGP
      IF (IBC.GT.NBCS) WRITE(6,490) TBCXP,TBCHGP,TBCTGP,TBCQGP,TBCPGP
488   FORMAT(1H+,65X,F6.2,2F8.1,F12.1,F8.1)
490   FORMAT(66X,F6.2,2F8.1,F12.1,F8.1)
499   CONTINUE
```

```
94    CONTINUE                                                          NINPRTT 2281
      DO 95 I = 1,200                                                   NINPRTT 2282
95    NFLUID(I) = 0                                                     NINPRTT 2283
100   WRITE(6,150) ICHNL                                                NINPRTT 2284
      IF (ICHNL.GT.1) GO TO 101                                         NINPRTT 2285
      IF (IHUB.EQ.1) WRITE(6,142)                                       NINPRTT 2286
      IF (IHUB.EQ.2) WRITE(6,144)                                       NINPRTT 2287
      IF (IHUB.EQ.3) WRITE(6,146)                                       NINPRTT 2288
101   CONTINUE                                                          NINPRTT 2289
      IF (ICHNL.LT.NSLICE) GO TO 102                                    NINPRTT 2290
      IF (ITIP.EQ.1) WRITE(6,147)                                       NINPRTT 2291
      IF (ITIP.EQ.2) WRITE(6,148)                                       NINPRTT 2292
      IF (ITIP.EQ.3) WRITE(6,149)                                       NINPRTT 2293
102   CONTINUE                                                          NINPRTT 2294
      TRIN = RIN(ICHNL)/CINCH(IUNITS)                                   NINPRTT 2295
      TROUT = ROUT(ICHNL)/CINCH(IUNITS)                                 NINPRTT 2296
      TDPLN = DPLN(ICHNL)/CINCH(IUNITS)                                 NINPRTT 2297
      TAPLN = APLN(ICHNL)/(CINCH(IUNITS)*CINCH(IUNITS))                 NINPRTT 2298
      WRITE(6,103) TRIN, UL(IUNITS),TROUT,UL(IUNITS),                   NINPRTT 2299
     Z             TDPLN,UL(IUNITS),TAPLN,UA(IUNITS)                    NINPRTT 2300
103   FORMAT(/' COOLANT PLENUM: RI=',F7.3,A4,'     RO=',F7.3,A4,4X,     NINPRTT 2301
     Z        'DHYD=',F7.4,A4,'     APLEN=',F7.4,A4,'*2')               NINPRTT 2302
C                                                                       NINPRTT 2303
      IF (IUNITS.EQ.1) GO TO 500                                        NINPRTT 2304
C                                                                       NINPRTT 2305
      WRITE(6,153) NFWD,NSTA,SPAN                                       NINPRTT 2306
      WRITE(6,155) CD,ADUMP                                             NINPRTT 2307
      TEM = TTIO(1) + 460.                                              NINPRTT 2308
      WRITE(6,157) TEM,PTIO(1),PEX(ICHNL),WPLEN                         NINPRTT 2309
      ITRBG = NFWD + 2                                                  NINPRTT 2310
      WRITE(6,154) ICHNL,ITRBG                                          NINPRTT 2311
      DO 118 I = 1,NSTA,20                                              NINPRTT 2312
      IP18 = I + 18                                                     NINPRTT 2313
      IF (IP18.GT.NSTA) IP18 = NSTA                                     NINPRTT 2314
      IF (I.EQ.1) WRITE(6,156) (J,J=I,IP18,2)                           NINPRTT 2315
      IF (I.GT.1) WRITE(6,159) (J,J=I,IP18,2)                           NINPRTT 2316
      ID = 0                                                            NINPRTT 2317
      DO 104 J = I,IP18,2                                               NINPRTT 2318
      ID = ID + 1                                                       NINPRTT 2319
      DUM1(ID) = RP(J)                                                  NINPRTT 2320
      NFLUID(J) = 5*J                                                   NINPRTT 2321
104   CONTINUE                                                          NINPRTT 2322
      WRITE(6,158) (NFLUID(J),J=I,IP18,2)                               NINPRTT 2323
      ID = 0                                                            NINPRTT 2324
      DO 116 J = I,IP18,2                                               NINPRTT 2325
      ID = ID + 1                                                       NINPRTT 2326
      NOS = NFLUID(J) - 4                                               NINPRTT 2327
      IF (NOS.GT.1) GO TO 106                                           NINPRTT 2328
      XOS = 0.0                                                         NINPRTT 2329
      XJN = 0.0                                                         NINPRTT 2330
      XMM = 0.0                                                         NINPRTT 2331
      XIS = 0.0                                                         NINPRTT 2332
      XCC = 0.0                                                         NINPRTT 2333
      GO TO 108                                                         NINPRTT 2334
106   XOS = XOS + DLX(NOS)                                              NINPRTT 2335
      XJN = XJN + DLX(NOS+1)                                            NINPRTT 2336
      XMM = XMM + DLX(NOS+2)                                            NINPRTT 2337
      XIS = XIS + DLX(NOS+3)                                            NINPRTT 2338
      XCC = XCC + DLX(NOS+4)                                            NINPRTT 2339
108   CONTINUE                                                          NINPRTT 2340
```

```
      DUM2(ID) = XOS
      DUM25(ID) = XJN
      DUM3(ID) = XMM
      DUM4(ID) = XIS
      DUM5(ID) = XCC
      DUM55(ID) = TAU(NOS)
      DUM6(ID) = TAU(NOS+2)
      NOS = NFLUID(J)
      DUM7(ID) = TAU(NOS)
      DUM8(ID) = A(NOS)
      DUM9(ID) = DH(J)
      DUM10(ID) = DHJ(J)
      DUM11(ID) = CNUM(J)
      DUM12(ID) = AJET(J)
      DUM16(ID) = THUBIN(J) - 460.
      DUM17(ID) = QHUBIN(J)
      DUM18(ID) = TTIPIN(J) - 460.
      DUM19(ID) = QTIPIN(J)
      NOS = NFLUID(J) - 4
      DUM13(ID) = TG(J) - 460.
      DUM14(ID) = HG(J)
      JHCAL = IHC(J)
      DUM15(ID) = HCAL(JHCAL)
      IF (BTA.GT..01) DUM14(ID) = QG(J)
116   CONTINUE
      WRITE(6,160) (DUM1(J),J=1,ID)
      WRITE(6,162) (DUM2(J),J=1,ID)
      WRITE(6,163) (DUM25(J),J=1,ID)
      WRITE(6,164) (DUM3(J),J=1,ID)
      WRITE(6,166) (DUM4(J),J=1,ID)
      WRITE(6,168) (DUM5(J),J=1,ID)
      WRITE(6,169) (DUM55(J),J=1,ID)
      WRITE(6,170) (DUM6(J),J=1,ID)
      WRITE(6,172) (DUM7(J),J=1,ID)
      WRITE(6,174) (DUM8(J),J=1,ID)
      WRITE(6,176) (DUM9(J),J=1,ID)
      WRITE(6,178) (DUM10(J),J=1,ID)
      WRITE(6,180) (DUM11(J),J=1,ID)
      WRITE(6,182) (DUM12(J),J=1,ID)
      WRITE(6,183) (DUM15(J),J=1,ID)
      WRITE(6,184) (DUM13(J),J=1,ID)
      IF (BTA.LT..01) WRITE(6,186) (DUM14(J),J=1,ID)
      IF (BTA.GT..01) WRITE(6,188) (DUM14(J),J=1,ID)
      IF (ICHNL.GT.1) GO TO 118
      IF (IHUB.EQ.1) WRITE(6,196) (DUM16(J),J=1,ID)
      IF (IHUB.EQ.3) WRITE(6,198) (DUM17(J),J=1,ID)
      IF (ITIP.EQ.1) WRITE(6,202) (DUM18(J),J=1,ID)
      IF (ITIP.EQ.3) WRITE(6,204) (DUM19(J),J=1,ID)
118   CONTINUE
      ITRBG = NFWD + 1
      WRITE(6,190) ICHNL,ITRBG
      XOS = 0.0
      XJN = 0.0
      XMM = 0.0
      XIS = 0.0
      XCC = 0.0
      DO 140 I = 2,NSTA,20
      IP18 = I + 18
      IF (IP18.GT.NSTA) IP18 = NSTA-1
      IF (I.EQ.2) WRITE(6,156) (J,J=I,IP18,2)
```

```
      IF (I.GT.2) WRITE(6,159) (J,J=I,IP18,2)                            NINPRTT 2401
      ID = 0                                                             NINPRTT 2402
      DO 122 J = I,IP18,2                                                NINPRTT 2403
      ID = ID + 1                                                        NINPRTT 2404
      DUM1(ID) = RR(J)                                                   NINPRTT 2405
      NFLUID(J) = 5*J                                                    NINPRTT 2406
122   CONTINUE                                                           NINPRTT 2407
      WRITE(6,158) (NFLUID(J),J=I,IP18,2)                                NINPRTT 2408
      ID = 0                                                             NINPRTT 2409
      DO 130 J = I,IP18,2                                                NINPRTT 2410
      ID = ID + 1                                                        NINPRTT 2411
      NOS = NFLUID(J) - 4                                                NINPRTT 2412
      XOS = XOS + DLX(NOS)                                               NINPRTT 2413
      XJN = XJN + DLX(NOS+1)                                             NINPRTT 2414
      XMM = XMM + DLX(NOS+2)                                             NINPRTT 2415
      XIS = XIS + DLX(NOS+3)                                             NINPRTT 2416
      XCC = XCC + DLX(NOS+4)                                             NINPRTT 2417
      DUM2(ID) = XOS                                                     NINPRTT 2418
      DUM25(ID) = XJN                                                    NINPRTT 2419
      DUM3(ID) = XMM                                                     NINPRTT 2420
      DUM4(ID) = XIS                                                     NINPRTT 2421
      DUM5(ID) = XCC                                                     NINPRTT 2422
      DUM55(ID) = TAU(NOS)                                               NINPRTT 2423
      DUM6(ID) = TAU(NOS+2)                                              NINPRTT 2424
      NOS = NFLUID(J)                                                    NINPRTT 2425
      DUM7(ID) = TAU(NOS)                                                NINPRTT 2426
      DUM8(ID) = A(NOS)                                                  NINPRTT 2427
      DUM9(ID) = DH(J)                                                   NINPRTT 2428
      DUM10(ID) = DHJ(J)                                                 NINPRTT 2429
      DUM11(ID) = CNUM(J)                                                NINPRTT 2430
      DUM12(ID) = AJET(J)                                                NINPRTT 2431
      DUM16(ID) = THUBIN(J) - 460.                                       NINPRTT 2432
      DUM17(ID) = QHUBIN(J)                                              NINPRTT 2433
      DUM18(ID) = TTIPIN(J) - 460.                                       NINPRTT 2434
      DUM19(ID) = QTIPIN(J)                                              NINPRTT 2435
      NOS = NFLUID(J) - 4                                                NINPRTT 2436
      DUM13(ID) = TG(J) - 460.                                           NINPRTT 2437
      DUM14(ID) = HG(J)                                                  NINPRTT 2438
      JHCAL = IHC(J)                                                     NINPRTT 2439
      DUM15(ID) = HCAL(JHCAL)                                            NINPRTT 2440
      IF (BTA.GT..01) DUM14(ID) = QG(J)                                  NINPRTT 2441
130   CONTINUE                                                           NINPRTT 2442
      WRITE(6,160) (DUM1(J),J=1,ID)                                      NINPRTT 2443
      WRITE(6,162) (DUM2(J),J=1,ID)                                      NINPRTT 2444
      WRITE(6,163) (DUM25(J),J=1,ID)                                     NINPRTT 2445
      WRITE(6,164) (DUM3(J),J=1,ID)                                      NINPRTT 2446
      WRITE(6,166) (DUM4(J),J=1,ID)                                      NINPRTT 2447
      WRITE(6,168) (DUM5(J),J=1,ID)                                      NINPRTT 2448
      WRITE(6,169) (DUM55(J),J=1,ID)                                     NINPRTT 2449
      WRITE(6,170) (DUM6(J),J=1,ID)                                      NINPRTT 2450
      WRITE(6,172) (DUM7(J),J=1,ID)                                      NINPRTT 2451
      WRITE(6,174) (DUM8(J),J=1,ID)                                      NINPRTT 2452
      WRITE(6,176) (DUM9(J),J=1,ID)                                      NINPRTT 2453
      WRITE(6,178) (DUM10(J),J=1,ID)                                     NINPRTT 2454
      WRITE(6,180) (DUM11(J),J=1,ID)                                     NINPRTT 2455
      WRITE(6,182) (DUM12(J),J=1,ID)                                     NINPRTT 2456
      WRITE(6,183) (DUM15(J),J=1,ID)                                     NINPRTT 2457
      WRITE(6,184) (DUM13(J),J=1,ID)                                     NINPRTT 2458
      IF (BTA.LT..01) WRITE(6,186) (DUM14(J),J=1,ID)                     NINPRTT 2459
      IF (BTA.GT..01) WRITE(6,188) (DUM14(J),J=1,ID)                     NINPRTT 2460
```

```
      IF (ICHNL.GT.1) GO TO 140                                         NINPRTT 2461
      IF (IHUB.EQ.1) WRITE(6,196)  (DUM16(J),J=1,ID)                    NINPRTT 2462
      IF (IHUB.EQ.3) WRITE(6,198)  (DUM17(J),J=1,ID)                    NINPRTT 2463
      IF (ITIP.EQ.1) WRITE(6,202)  (DUM18(J),J=1,ID)                    NINPRTT 2464
      IF (ITIP.EQ.3) WRITE(6,204)  (DUM19(J),J=1,ID)                    NINPRTT 2465
  140 CONTINUE                                                          NINPRTT 2466
  150 FORMAT(1H1,46X,'INPUT FOR SLICE NUMBER',I3)                       NINPRTT 2467
  142 FORMAT(21X,'  HUB TEMPERATURES ARE SPECIFIED')                    NINPRTT 2468
  144 FORMAT(21X,'  ADIABATIC HUB SPECIFIED')                           NINPRTT 2469
  146 FORMAT(21X,'  HUB HEAT FLUX IS SPECIFIED')                        NINPRTT 2470
  147 FORMAT(21X,'  TIP TEMPERATURES ARE SPECIFIED')                    NINPRTT 2471
  148 FORMAT(21X,'  ADIABATIC TIP SPECIFIED')                           NINPRTT 2472
  149 FORMAT(21X,'  TIP HEAT FLUX IS SPECIFIED')                        NINPRTT 2473
  153 FORMAT(/' NUMBER OF STATIONS IN IMPINGEMENT REGION IS',I3,        NINPRTT 2474
     Z                               ', TOTAL NUMBER OF STATIONS IS',I3, NINPRTT 2475
     Z       ', SPAN OF THIS SLICE IS',F6.3,' IN')                      NINPRTT 2476
  155 FORMAT(' IMPINGEMENT HOLE DISCHARGE COEF.=',F6.3,                 NINPRTT 2477
     Z  ',              AREA OF DUMP TO TRAILING EDGE =',F8.5,' IN**2')  NINPRTT 2478
  157 FORMAT(' COOLANT INLET TEMP.=',F7.1,' R,  COOLANT INLET PRESSURE',NINPRTT 2479
     Z                  ' =',F6.1,' PSIA,  EXIT PRESSURE =',            NINPRTT 2480
     Z      F6.1,' PSIA,'/' COOLANT FLOW =',F6.1,' LBM/HR')             NINPRTT 2481
  154 FORMAT(/' PRESSURE SIDE, SLICE ',I2,', TRAILING EDGE REGION ',    NINPRTT 2482
     Z       'BEGINS AT STATION-',I3)                                   NINPRTT 2483
  156 FORMAT(/' STATION NUMBER',5X,10(6X,I4))                           NINPRTT 2484
  158 FORMAT(' COOLANT NODE NUMBER',10(6X,I4))                          NINPRTT 2485
  159 FORMAT(1H2//' STATION NUMBER',5X,10(6X,I4))                       NINPRTT 2486
  160 FORMAT(' RADIAL LOCATION(IN)',10F10.3)                            NINPRTT 2487
  162 FORMAT(' X, OUTSIDE SUR.(IN)',10F10.5)                            NINPRTT 2488
  163 FORMAT(' X, INTERFACE   (IN)',10F10.5)                            NINPRTT 2489
  164 FORMAT(' X, MID-METAL   (IN)',10F10.5)                            NINPRTT 2490
  166 FORMAT(' X, INSIDE SURF.(IN)',10F10.5)                            NINPRTT 2491
  168 FORMAT(' X, MID.COOL.CH.(IN)',10F10.5)                            NINPRTT 2492
  169 FORMAT(/' COATING THKNSS (IN)',10F10.5)                           NINPRTT 2493
  170 FORMAT(' WALL THICKNESS (IN)',10F10.5)                            NINPRTT 2494
  172 FORMAT(' CHANNEL WIDTH  (IN)',10F10.5)                            NINPRTT 2495
  174 FORMAT(' CHANNEL AREA(IN**2)',10F10.5)                            NINPRTT 2496
  176 FORMAT(' CHANNEL HYD.DIA(IN)',10F10.5)                            NINPRTT 2497
  178 FORMAT(/' IMP.JET HYD.DIA(IN)',10F10.5)                           NINPRTT 2498
  180 FORMAT(' NO. OF IMP. JETS   ',10F10.2)                            NINPRTT 2499
  182 FORMAT(' TOT.JET AREA(IN**2)',10F10.5)                            NINPRTT 2500
  183 FORMAT(/' TYPE OF HC CALC.   ',10(6X,A4))                         NINPRTT 2501
  184 FORMAT(' OUTSIDE BC:  TG,(F)',10F10.1)                            NINPRTT 2502
  186 FORMAT(' HG (BTU/HR/FT**2/R)',10F10.1)                            NINPRTT 2503
  188 FORMAT(' QG   (BTU/HR/FT**2)',10F10.1)                            NINPRTT 2504
  190 FORMAT(1H1,/' SUCTION SIDE, SLICE ',I2,', TRAILING EDGE REGION',  NINPRTT 2505
     Z         ' BEGINS AT STATION-',I3)                                NINPRTT 2506
  192 FORMAT(/' CLAD K(BTU/HR/FT/R)',10F10.3)                           NINPRTT 2507
  194 FORMAT(' METL K(BTU/HR/FT/R)',10F10.3)                            NINPRTT 2508
  196 FORMAT(' GIVEN HUB TEMP. (F)',10F10.1)                            NINPRTT 2509
  198 FORMAT(' QHUB (BTU/HR/FT**2)',10F10.1)                            NINPRTT 2510
  202 FORMAT(' GIVEN TIP TEMP. (F)',10F10.1)                            NINPRTT 2511
  204 FORMAT(' QTIP (BTU/HR/FT**2)',10F10.1)                            NINPRTT 2512
C                                                                       NINPRTT 2513
      GO TO 350                                                         NINPRTT 2514
C                                                                       NINPRTT 2515
  500 SPANC = SPAN/CINCH(1)                                             NINPRTT 2516
      ADUMPC = ADUMP/(CINCH(1)**2)                                      NINPRTT 2517
      WRITE(6,553) NFWD,NSTA,SPANC                                      NINPRTT 2518
      WRITE(6,555) CD,ADUMPC                                            NINPRTT 2519
      TEM = (TTIO(1) + 460.)/1.8                                        NINPRTT 2520
```

```
      PTIOC = PTIO(1)/CPRSR(1)
      PEXC = PEX(ICHNL)/CPRSR(1)
      WPLENC = WPLEN/CMSFL(1)
      WRITE(6,557) TEM,PTIOC,PEXC,WPLENC
      ITRBG = NFWD + 2
      WRITE(6,154) ICHNL,ITRBG
      DO 518 I = 1,NSTA,20
      IP18 = I + 18
      IF (IP18.GT.NSTA) IP18 = NSTA
      IF (I.EQ.1) WRITE(6,556) (J,J=I,IP18,2)
      IF (I.GT.1) WRITE(6,559) (J,J=I,IP18,2)
      ID = 0
      DO 504 J = I,IP18,2
      ID = ID + 1
      DUM1(ID) = RR(J)/CINCH(1)
      NFLUID(J) = 5*J
  504 CONTINUE
      WRITE(6,558) (NFLUID(J),J=I,IP18,2)
      ID = 0
      DO 516 J = I,IP18,2
      ID = ID + 1
      NOS = NFLUID(J) - 4
      IF (NOS.GT.1) GO TO 506
      XOS = 0.0
      XJN = 0.0
      XMM = 0.0
      XIS = 0.0
      XCC = 0.0
      GO TO 508
  506 XOS = XOS + DLX(NOS)/CINCH(1)
      XJN = XJN + DLX(NOS+1)/CINCH(1)
      XMM = XMM + DLX(NOS+2)/CINCH(1)
      XIS = XIS + DLX(NOS+3)/CINCH(1)
      XCC = XCC + DLX(NOS+4)/CINCH(1)
  508 CONTINUE
      DUM2(ID) = XOS
      DUM25(ID) = XJN
      DUM3(ID) = XMM
      DUM4(ID) = XIS
      DUM5(ID) = XCC
      DUM55(ID) = TAU(NOS)/CINCH(1)
      DUM6(ID) = TAU(NOS+2)/CINCH(1)
      NOS = NFLUID(J)
      DUM7(ID) = TAU(NOS)/CINCH(1)
      DUM8(ID) = A(NOS)/(CINCH(1)**2)
      DUM9(ID) = DH(J)/CINCH(1)
      DUM10(ID) = DHJ(J)/CINCH(1)
      DUM11(ID) = CNUM(J)
      DUM12(ID) = AJET(J)/(CINCH(1)**2)
      DUM16(ID) = THUBIN(J)/1.8
      DUM17(ID) = QHUBIN(J)/CHFLX(1)
      DUM18(ID) = TTIPIN(J)/1.8
      DUM19(ID) = QTIPIN(J)/CHFLX(1)
      NOS = NFLUID(J) - 4
      DUM13(ID) = TG(J)/1.8
      DUM14(ID) = HG(J)/CHTC(1)
      JHCAL = IHC(J)
      DUM15(ID) = HCAL(JHCAL)
      IF (BTA.GT..01) DUM14(ID) = QG(J)/CHFLX(1)
  516 CONTINUE
```

```
      WRITE(6,560)  (DUM1(J),J=1,ID)
      WRITE(6,562)  (DUM2(J),J=1,ID)
      WRITE(6,563)  (DUM25(J),J=1,ID)
      WRITE(6,564)  (DUM3(J),J=1,ID)
      WRITE(6,566)  (DUM4(J),J=1,ID)
      WRITE(6,568)  (DUM5(J),J=1,ID)
      WRITE(6,569)  (DUM55(J),J=1,ID)
      WRITE(6,570)  (DUM6(J),J=1,ID)
      WRITE(6,572)  (DUM7(J),J=1,ID)
      WRITE(6,574)  (DUM8(J),J=1,ID)
      WRITE(6,576)  (DUM9(J),J=1,ID)
      WRITE(6,578)  (DUM10(J),J=1,ID)
      WRITE(6,580)  (DUM11(J),J=1,ID)
      WRITE(6,582)  (DUM12(J),J=1,ID)
      WRITE(6,583)  (DUM15(J),J=1,ID)
      WRITE(6,584)  (DUM13(J),J=1,ID)
      IF (BTA.LT..01) WRITE(6,586)  (DUM14(J),J=1,ID)
      IF (BTA.GT..01) WRITE(6,588)  (DUM14(J),J=1,ID)
      IF (ICHNL.GT.1) GO TO 518
      IF (IHUB.EQ.1) WRITE(6,596)  (DUM16(J),J=1,ID)
      IF (IHUB.EQ.3) WRITE(6,598)  (DUM17(J),J=1,ID)
      IF (ITIP.EQ.1) WRITE(6,602)  (DUM18(J),J=1,ID)
      IF (ITIP.EQ.3) WRITE(6,604)  (DUM19(J),J=1,ID)
  518 CONTINUE
      ITRBG = NFWD + 1
      WRITE(6,190) ICHNL,ITRBG
      XOS = 0.0
      XJN = 0.0
      XMM = 0.0
      XIS = 0.0
      XCC = 0.0
      DO 540 I = 2,NSTA,20
      IP18 = I + 18
      IF (IP18.GT.NSTA) IP18 = NSTA-1
      IF (I.EQ.2) WRITE(6,556) (J,J=I,IP18,2)
      IF (I.GT.2) WRITE(6,559) (J,J=I,IP18,2)
      ID = 0
      DO 522 J = I,IP18,2
      ID = ID + 1
      DUM1(ID) = RR(J)/CINCH(1)
      NFLUID(J) = 5*J
  522 CONTINUE
      WRITE(6,558) (NFLUID(J),J=I,IP18,2)
      ID = 0
      DO 530 J = I,IP18,2
      ID = ID + 1
      NOS = NFLUID(J) - 4
      XOS = XOS + DLX(NOS)/CINCH(1)
      XJN = XJN + DLX(NOS+1)/CINCH(1)
      XMM = XMM + DLX(NOS+2)/CINCH(1)
      XIS = XIS + DLX(NOS+3)/CINCH(1)
      XCC = XCC + DLX(NOS+4)/CINCH(1)
      DUM2(ID) = XOS
      DUM25(ID) = XJN
      DUM3(ID) = XMM
      DUM4(ID) = XIS
      DUM5(ID) = XCC
      DUM55(ID) = TAU(NOS)/CINCH(1)
      DUM6(ID) = TAU(NOS+2)/CINCH(1)
      NOS = NFLUID(J)
```

```
      DUM7(ID)  = TAU(NOS)/CINCH(1)
      DUM8(ID)  = A(NOS)/(CINCH(1)**2)
      DUM9(ID)  = DH(J)/CINCH(1)
      DUM10(ID)  = DHJ(J)/CINCH(1)
      DUM11(ID)  = CNUM(J)
      DUM12(ID)  = AJET(J)/(CINCH(1)**2)
      DUM16(ID)  = THUBIN(J)/1.8
      DUM17(ID)  = QHUBIN(J)/CHFLX(1)
      DUM18(ID)  = TTIPIN(J)/1.8
      DUM19(ID)  = QTIPIN(J)/CHFLX(1)
      NOS = NFLUID(J) - 4
      DUM13(ID)  = TG(J)/1.8
      DUM14(ID)  = HG(J)/CHTC(1)
      JHCAL = IHC(J)
      DUM15(ID)  = HCAL(JHCAL)
      IF (BTA.GT..01) DUM14(ID)  = QG(J)/CHFLX(1)
530   CONTINUE
      WRITE(6,560)  (DUM1(J),J=1,ID)
      WRITE(6,562)  (DUM2(J),J=1,ID)
      WRITE(6,563)  (DUM25(J),J=1,ID)
      WRITE(6,564)  (DUM3(J),J=1,ID)
      WRITE(6,566)  (DUM4(J),J=1,ID)
      WRITE(6,568)  (DUM5(J),J=1,ID)
      WRITE(6,569)  (DUM55(J),J=1,ID)
      WRITE(6,570)  (DUM6(J),J=1,ID)
      WRITE(6,572)  (DUM7(J),J=1,ID)
      WRITE(6,574)  (DUM8(J),J=1,ID)
      WRITE(6,576)  (DUM9(J),J=1,ID)
      WRITE(6,578)  (DUM10(J),J=1,ID)
      WRITE(6,580)  (DUM11(J),J=1,ID)
      WRITE(6,582)  (DUM12(J),J=1,ID)
      WRITE(6,583)  (DUM15(J),J=1,ID)
      WRITE(6,584)  (DUM13(J),J=1,ID)
      IF (BTA.LT..01) WRITE(6,586)  (DUM14(J),J=1,ID)
      IF (BTA.GT..01) WRITE(6,588)  (DUM14(J),J=1,ID)
      IF (ICHNL.GT.1) GO TO 540
      IF (IHUB.EQ.1) WRITE(6,596)  (DUM16(J),J=1,ID)
      IF (IHUB.EQ.3) WRITE(6,598)  (DUM17(J),J=1,ID)
      IF (ITIP.EQ.1) WRITE(6,602)  (DUM18(J),J=1,ID)
      IF (ITIP.EQ.3) WRITE(6,604)  (DUM19(J),J=1,ID)
540   CONTINUE
553   FORMAT(/' NUMBER OF STATIONS IN IMPINGEMENT REGION IS',I3,
     Z         ', TOTAL NUMBER OF STATIONS IS',I3,
     Z       ', SPAN OF THIS SLICE IS',F6.3,' CM')
555   FORMAT(' IMPINGEMENT HOLE DISCHARGE COEF.=',F6.3,
     Z    ',             AREA OF DUMP TO TRAILING EDGE =',F8.5,' CM**2')
557   FORMAT(' COOLANT INLET TEMP.=',F7.1,' K,  COOLANT INLET',
     Z          ' PRESSURE =',F7.1,' KPA,  EXIT PRESSURE =',
     Z       F7.1,' KPA,'/' COOLANT FLOW =',F6.1,' KG/HR')
556   FORMAT(//' STATION NUMBER',5X,10(6X,I4))
558   FORMAT(' COOLANT NODE NUMBER',10(6X,I4))
559   FORMAT(1H2//' STATION NUMBER',5X,10(6X,I4))
560   FORMAT(' RADIAL LOCATION(CM)',10F10.3)
562   FORMAT(' X, OUTSIDE SUR.(CM)',10F10.5)
563   FORMAT(' X, INTERFACE   (CM)',10F10.5)
564   FORMAT(' X, MID-METAL   (CM)',10F10.5)
566   FORMAT(' X, INSIDE SURF.(CM)',10F10.5)
568   FORMAT(' X, MID.COOL.CH.(CM)',10F10.5)
569   FORMAT(/' COATING THKNSS (CM)',10F10.5)
570   FORMAT(' WALL THICKNESS (CM)',10F10.5)
```

```
572    FORMAT(' CHANNEL WIDTH  (CM)',10F10.5)                          NINPRTT 2701
574    FORMAT(' CHANNEL AREA(CM**2)',10F10.5)                          NINPRTT 2702
576    FORMAT(' CHANNEL HYD.DIA(CM)',10F10.5)                          NINPRTT 2703
578    FORMAT(/' IMP.JET HYD.DIA(CM)',10F10.5)                         NINPRTT 2704
580    FORMAT(' NO. OF IMP. JETS   ',10F10.2)                          NINPRTT 2705
582    FORMAT(' TOT.JET AREA(CM**2)',10F10.5)                          NINPRTT 2706
583    FORMAT(/' TYPE OF HC CALC.   ',10(6X,A4))                       NINPRTT 2707
584    FORMAT(' OUTSIDE BC:  TG,(K)',10F10.1)                          NINPRTT 2708
586    FORMAT(' HG        (W/M**2/K)',10F10.1)                         NINPRTT 2709
588    FORMAT(' QG           (W/M**2)',10F10.1)                        NINPRTT 2710
592    FORMAT(/' CLAD K       (W/M/K)',10F10.3)                        NINPRTT 2711
594    FORMAT(' METAL K       (W/M/K)',10F10.3)                        NINPRTT 2712
596    FORMAT(' GIVEN HUB TEMP. (K)',10F10.1)                          NINPRTT 2713
598    FORMAT(' QHUB          (W/M**2)',10F10.1)                       NINPRTT 2714
602    FORMAT(' GIVEN TIP TEMP. (K)',10F10.1)                          NINPRTT 2715
604    FORMAT(' QTIP          (W/M**2)',10F10.1)                       NINPRTT 2716
C                                                                      NINPRTT 2717
350    CONTINUE                                                        NINPRTT 2718
       RETURN                                                          NINPRTT 2719
       END                                                             NINPRTT 2720


C----SOURCE.NPARAYT                                                    NPARAYT 2721
      SUBROUTINE PARRAY(JS,JSENS,ICHOKE)                               NPARAYT 2722
C                                                                      NPARAYT 2723
C-    SOURCE.NPARAYT----                                               NPARAYT 2724
C+++ A SUBROUTINE TO SET UP THE COEF ARRAY TO SOLVE FOR BLADE PRESSURES NPARAYT 2725
C                                                                      NPARAYT 2726
      REAL*8 TCOF                                                      NPARAYT 2727
C                                                                      NPARAYT 2728
      COMMON /MATRX/ TCOF(400,30)                                      NPARAYT 2729
C                                                                      NPARAYT 2730
      COMMON /PRPS/ CPO,       GAMO,      DP(80),     SP(80),  RE(80), NPARAYT 2731
     Z              CPC(80),   GAMC(80),  DUMR1(80),  DUMR2(80)        NPARAYT 2732
C                                                                      NPARAYT 2733
      COMMON /TCO/ ADUMP,      BTA,       CD,         CP,              NPARAYT 2734
     Z             GAM,        PIM,       R,          SPAN,       TOG, NPARAYT 2735
     Z             WDUMP,      WIM,       AKC(15,80), AKW(15,80),      NPARAYT 2736
     Z             A(400),     AJET(80),  AM2(80),    CNUM(80),        NPARAYT 2737
     Z             DH(80),     DHF(80),   DHJ(80),                     NPARAYT 2738
     Z             DLX(400),   FF(80),    HC(80),     HG(80),          NPARAYT 2739
     Z             P(2,15,80),PEXIT(15),  PUMP(80),   QG(80),          NPARAYT 2740
     Z             QSNK(80),   RR(80),    S(15),      T(2,15,400),     NPARAYT 2741
     Z             TG(80),     TAU(400),  WFC(80),                     NPARAYT 2742
     Z             WJ(15,80),  WCROS(2,15,80),        XN(80),          NPARAYT 2743
     Z             ICOR,       IFILM,     IHUB,       ITIP,            NPARAYT 2744
     Z             ISBLOK,     ISLICE,    NBLKSZ,     NSLICE,          NPARAYT 2745
     Z             NFWD,       NSTA,      IHC(80)                      NPARAYT 2746
C                                                                      NPARAYT 2747
      COMMON /TRNSNT/ RHOC,       RHOM,       SPHTC,      SPHTM,       NPARAYT 2748
     Z                DLTYME,     TYME,       TEPS,       TYMMAX       NPARAYT 2749
C                                                                      NPARAYT 2750
      DIMENSION POLD(80), PSAV(5)                                      NPARAYT 2751
C                                                                      NPARAYT 2752
C COMPUTE NEW PRESSURES                                                NPARAYT 2753
C                                                                      NPARAYT 2754
C IFNL = THE NUMBER OF FLOW CHANNEL NODES                              NPARAYT 2755
C                                                                      NPARAYT 2756
      TREPS = 1.0                                                      NPARAYT 2757
      IF (TYME.GE.0.) TREPS = TEPS                                     NPARAYT 2758
800   IFNL = NSTA - 3                                                  NPARAYT 2759
      NODST = 5*NSTA                                                   NPARAYT 2760
```

```
      NODSF = 5*NFWD
C
C INITIALIZE COEFFICIENT ARRAY TO 0.0
C
      DO 810 I = 1,IFNL
      DO 810 J = 1,30
810   TCOF(I,J) = 0.0
C
C COMPUTE THE COEFFICIENT VALUES
C
      DO 900 I = 1,IFNL
      FILM = 0.0
820   ICHK = I - 2*(I/2)
C
C FOR THE IMPINGEMENT REGION:
C     ICHK = 0 IMPLIES I IS EVEN AND STATION IS ON SUCTION SIDE
C          = 1 IMPLIES THAT I IS ODD AND STATION IS ON PRESSURE SIDE
C
C DEFINE THE REAL NODE NUMBER IN TERMS OF I
C WHERE IRL IS THE PIVOTAL ELEMENT = COOLANT NODE NUMBER, LCOOL
C        IDN = DOWNSTREAM COOLANT NODE
C        IUP = UPSTREAM COOLANT NODE
C
      IF ( I.LT.NFWD ) GO TO 840
      IF ( I.EQ.NFWD) GO TO 890
C
C FOR I=NFWD, THE NODE IS THE ENTRANCE TO THE TRAILING EDGE AND IS
C  TREATED SEPARATELY AT (890)
C FOR I>NFWD, THE NODE IS IN THE TRAILING EDGE AND IRL IS DEFINED AS:
C
      IF (ICHK.GT.0) GO TO 885
      IRL  = 5*I
      IDN  = IRL + 10
      IDNS = I+2
      IUP  = IRL
      IUPS = IDNS - 2
      ITC  = 10
      ITCP = 12
830   CONTINUE
      GO TO 860
C
840   CONTINUE
      IRL = 5*I
      IF (I.GT.JS) GO TO 843
C
      IF (I.LT.JS) GO TO 852
C
      IF (ICHK.GT.0) GO TO 849
C
      GO TO 855
C
843   IF (ICHK.GT.0) GO TO 849
C
C  STATION I IS SUCTION SIDE, DOWNSTREAM OF SPLIT POINT
C
846   IUPS = I - 2
      IDNS = I
      IUP  = IRL - 10
      IDN  = IRL
      ITC  = 8
```

```
      ITCP = 10
      IF (I.GT.2) GO TO 860
      IUPS = 1
      IUP  = 5
      ITC  = 9
      GO TO 860
C
C  STATION I IS PRESSURE SIDE,DOWNSTREAM OF SPLIT POINT
C
849   CONTINUE
      IUPS = I
      IDNS = I + 2
      IUP  = IRL
      IDN  = IRL + 10
      ITC  = 10
      ITCP = 12
      GO TO 860
852   CONTINUE
      IF (ICHK.GT.0) GO TO 858
      IF (ICHK.NE.JSENS) GO TO 846
855   CONTINUE
C
C  I IS ON SUCTION SIDE, FORWARD OF SPLIT POINT
C
      IUPS = I
      IDNS = I - 2
      IUP  = IRL
      IDN  = IRL - 10
      IDX = IUP
      ITC  = 10
      ITCP = 8
      IF (I.GT.2) GO TO 860
      IDNS = 1
      IDN  = 5
      ITCP = 9
      GO TO 860
858   CONTINUE
      IF (ICHK.NE.JSENS) GO TO 849
      IDNS = I
      IUPS = I + 2
      IDN  = IRL
      IUP  = IRL + 10
      IDX = IUP
      ITC  = 12
      ITCP = 10
C
860   CONTINUE
C
C
      TRTRM = 0.0
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRM = 12.*DLX(IDN) *
     Z (WCROS(2,ISLICE,IDNS)-WCROS(1,ISLICE,IDNS))/(DLTYME*A(IUP)*32.2)
      WFCDUM = WFC(IDNS)
      IF (I.GT.NFWD) WFCDUM = WFCDUM + WFC(IDNS+1)
      IF(WCROS(2,ISLICE,IDNS).NE.0.0) FILM = WFCDUM/WCROS(2,ISLICE,IDNS)
      TCOF(I,ITC)   = TREPS*
     Z      ((1.0 + GAMC(IUPS)*AM2(IUPS))  + (A(IDN)-A(IUP))/(2.*A(IUP)))
      TCOF(I,ITCP) = TREPS*(-(1.0 + .5*GAMC(IDNS)*AM2(IDNS)*
     Z       (4.*FF(IDNS)*DLX(IDX)/DH(IDNS)+2.+2.*FILM))*A(IDN)/A(IUP)
     Z                 + (A(IDN)-A(IUP))/(2.*A(IUP)))
```

```
      ROOT           = SQRT(32.2*GAMC(IDNS)*R*T(2,ISLICE,IDN)*AM2(IDNS))
      PUMTRM = 0.0
      IF (ROOT.NE.0.0) PUMTRM = (3.14159265*WS/30.)**2*
     Z    RR(IDNS)*(RR(IDNS)-RR(IUPS))*WCROS(2,ISLICE,IDNS)
     Z                         /(A(IUP)*ROOT*144.*32.2)
      TCOF(I,20)     = -PUMTRM   + TRTRM - (1.-TREPS)*
     Z   (P(1,ISLICE,IUPS)*TCOF(I,ITC)  + P(1,ISLICE,IDNS)*TCOF(I,ITCP))
870   CONTINUE
      IF (IDNS.NE.ICHOKE) GO TO 880
      TCOF(I,20)  = -P(1,ISLICE,ICHOKE)*TCOF(I,12)  + TCOF(I,20)
      TCOF(I,12)  = 0.0
880   CONTINUE
C
C FOR TRAILING EDGE CHANNELS:
C
      IF (I.LT.IFNL) GO TO 900
C
C TCOF(I,20) IS NON-ZERO ONLY FOR I=IFNL
C
      IF (ICHOKE.EQ.NSTA-1) GO TO 900
      TCOF(I,20)  = - PEXIT(ISLICE)*TCOF(I,12)  + TCOF(I,20)
      TCOF(I,12)  = 0.0
      GO TO 900
885   CONTINUE
C
C  FOR A PRESSURE SIDE, TRAILING EDGE REGION STATION, COOLANT NODE
C   IS IDENTICAL TO SUCTION SIDE NODE.
C
      TCOF(I,10)  = 1.0
      TCOF(I,9)   =-1.0
      TCOF(I,20)  = 0.0
      GO TO 900
890   CONTINUE
C
C FOR THE SPECIAL NODE AT THE ENTRANCE TO THE TRAILING EDGE:
C ALLOWING FOR THE POSSIBILITY OF ADDITION OF EXTRA COOLING AIR
C  INTO TRAILING EDGE,
C
C
      TRTRM = 0.0
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRM=12.*DLX(NFWD+1)*
     Z     (WCROS(2,ISLICE,NFWD+1)-WCROS(1,ISLICE,NFWD+1))/
     Z         (DLTYME*A(NODSF+5)*32.2)
      AVRGA = (A(NODSF-5) + A(NODSF) - A(NODSF+5))/(3.*A(NODSF+5))
      TCOF(I,9)      = TREPS*((1. + GAMC(NFWD-1)*AM2(NFWD-1))*
     Z                 A(NODSF-5)/A(NODSF+5) - AVRGA)
      TCOF(I,10)     = TREPS*((1. + GAMC(NFWD)*AM2(NFWD))*
     Z                 A(NODSF)/A(NODSF+5)  - AVRGA)
      IF (WCROS(2,ISLICE,NFWD+1).NE.0.0) FILM =
     Z         (WFC(NFWD+1)+WFC(NFWD+2))/WCROS(2,ISLICE,NFWD+1)
      TCOF(I,11)     = TREPS*(-1. - GAMC(NFWD+1)*AM2(NFWD+1)*
     Z     ( 1. + 2.*FF(NFWD+1)*DLX(NODSF+5)/DH(NFWD +1)+FILM)  - AVRGA)
C
      PUMP(NFWD+1)  = (3.14159265*WS/30.)**2*
     Z                  RR(NFWD+1)*(RR(NFWD+1)-RR(NFWD))
      ROOT           = SQRT(32.2*GAMC(NFWD+1)*R*
     Z                      T(2,ISLICE,NODSF+5)*AM2(NFWD+1))
      PUMTRM = 0.0
      IF (ROOT.NE.0.0) PUMTRM = (3.14159265*WS/30.)**2*RR(NFWD+1)*
     Z      (RR(NFWD+1)-RR(NFWD))*WCROS(2,ISLICE,NFWD+1)
```

```
     Z                          /(A(NODSF+5)*ROOT*144.*32.2)
C
C  PUMP HAS UNITS OF (IN**2/SEC**2);   ROOT HAS UNITS OF (FT/SEC)
C
      DUMTER = 0.0
      IF (ADUMP.GT.0.) DUMTER = - WDUMP**2*R*
     Z                 (T(2,ISLICE,NODSF-5)+T(2,ISLICE,NODSF))/
     Z ((P(1,ISLICE,NFWD-1) + P(1,ISLICE,NFWD))*ADUMP*A(NODSF+5)*32.2)
      TCOF(I,20) = - PUMTRM + DUMTER + TRTRM
     Z  - (1.0-TREPS)*(P(1,ISLICE,NFWD-1)*TCOF(I,9)
     Z    + P(1,ISLICE,NFWD)*TCOF(I,10)+P(1,ISLICE,NFWD+1)*TCOF(I,11))
900   CONTINUE
C
      RETURN
      END


C----SOURCE.NPLENMP
      SUBROUTINE PLNUM(WXX,PXX,PTEXIT,TXX,TTEXIT)
C
C-    SOURCE.NPLENMP----
C     A SUBROUTINE TO COMPUTE PRESSURE DROP IN THE CENTRAL COOLANT PLENUM
C
C
C
C
C        ---------------------------------------------------------------
C      -
C        ARGUMENTS FOR THIS SUBROUTINE ARE:
C
C        WXX    = MASS FLOW RATE INTO THIS SLICE, LBM/HR
C        PXX    = AVERAGE STATIC PRESSURE FOR THIS PLENUM SLICE, PSIA,
C                       CALCULATED IN THIS SUBROUTINE.
C        PTEXIT = TOTAL PRESSURE IN AND TOTAL PRESSURE OUT, PSIA
C        TXX    = AVERAGE STATIC TEMPERATURE FOR THIS PLENUM SLICE, (F),
C                   CALCULATED IN THIS SUBROUTINE
C        TTEXIT = TOTAL TEMP. IN AND TOTAL TEMP.OUT, (F)
C
C        ---------------------------------------------------------------
C
C
      COMMON /RADL/ APLN(15),  DPLN(15),   RIN(15),    ROUT(15),
     Z              PIN(15),   TIN(15),    W(15),      WS
C
      COMMON /TCO/ ADUMP,      BTA,        CD,         CP,
     Z             GAM,        PIM,        R,          SPAN,       TOG,
     Z             WDUMP,      WIM,        AKC(15,80), AKW(15,80),
     Z             A(400),     AJET(80),   AM2(80),    CNUM(80),
     Z             DH(80),     DHF(80),    DHJ(80),
     Z             DLX(400),   FF(80),     HC(80),     HG(80),
     Z             P(2,15,80),PEXIT(15),   PUMP(80),   QG(80),
     Z             QSNK(80),   RR(80),     S(15),      T(2,15,400),
     Z             TG(80),     TAU(400),   WFC(80),
     Z             WJ(15,80),  WCROS(2,15,80),         XN(80),
     Z             ICOR,       IFILM,      IHUB,       ITIP,
     Z             ISBLOK,     ISLICE,     NBLKSZ,     NSLICE,
     Z             NFWD,       NSTA,       IHC(80)
C
      COMMON /TRNSNT/ RHOC,       RHOM,       SPHTC,      SPHTM,
     Z                DLTYME,     TYME,       TEPS,       TYMMAX
C
      COMMON /UNITS/ CINCH(2), CHTC(2),  CHFLX(2), CPRSR(2), CMSFL(2),
```

```
     Z                    CTMPF(2), CTCON(2), CDEN(2),  CSPHT(2), CGASC(2),
     Z                    CVISC(2), CRHOVG(2),IUNITS
C
      DIMENSION BETTA(20),B(20),AMC(20),SIGMA(20),TT1(20),F1(20),
     1 Z1(15),CH(15)
      DIMENSION SV(3),XK(4),XL(4)
C
C----FUNP(FF) IS THE EQUATION FOR DELTA P OVER LENGTH DX
C
      FUNP(FF)=(D1*RRP*(PP/R/TP-V1)/144.0-2.*FF*G2*TP/PP/AA/AA/DD)
     Z          /(1.-TP*G2/(PP*AA)**2+G1*CP
     Z          *778.161*R*TP*TP*V1/(PP*AA)**2/PP)*DX
C
C----FUNT(XK) IS THE EQUATION FOR DELTA T OVER LENGTH DX
C
      FUNT(XK)=(D2*RRP/CP+G1*R*(TP/PP/AA)**2*(XK/PP/DX))
     Z         /(1.+G1*R*TP/(PP*AA)**2)*DX
C
C     INITIALIZE
1     CONTINUE
      DIFTOL=0.005
      ACH=1.
      KSIG=0
      NCC=1
      IS=0
      KTR1=0
      W(ISLICE) = WXX
C
C   SAVE INLET TOTAL PRESSURE (PSIA) IN PIN AND INLET TOTAL
C     TEMPERATURE (F) IN TIN
C
      PIN(ISLICE) = PTEXIT
      TIN(ISLICE) = TTEXIT
      CH(ISLICE)=0.0
      ZED=Z1(ISLICE)*1.01
      IF (ZED.EQ.0.) ZED=.001
      IF (TIN(ISLICE).GT.-430.0) GO TO 5
3     TIN(ISLICE)=50.0
5     SIGB=0.0
C
C
7     NSTNS= 4
      SEGMTS=NSTNS-1
C
C
      T1=TIN(ISLICE)+460.0
      B(1)=T1
      BETA1=PIN(ISLICE)**2
      BETTA(1)=BETA1
      DX=S(ISLICE)/SEGMTS
      DXTEMP=DX
      XXN=NSTNS
C
C
13    DR=(ROUT(ISLICE)-RIN(ISLICE))/SEGMTS
C
CCOMPUTE CONSTANT TERMS-C1-C8 -

17    CONTINUE
      TTX=B(1)
```

```
      CALL GASTBL(TTX,C,CP,GAM,PD,R,XMU)
      J=1
      C6=.5*(GAM-1.0)
      C1=GAM/C6
C
C
      IF (WS)21,19,21
C NO PUMPING
19    C3=0.0
      GO TO 23
C
C   PUMPING
C
21    C3=2.36695E-6*(WS**2)/(C1*R)
23    C8=32.17*GAM*R
      C5=1.0/SQRT(C8)
      C7=1.0/(32.17*C1*R)
      IF (J.GT.1) GO TO 33
25    CONTINUE
C
C   COMPUTE CHANNEL REYNOLDS NO.  IF J = 1
C
      REY = 12.0*W(ISLICE)/XMU*DPLN(ISLICE)/APLN(ISLICE)
C
C   COMPUTE FRICTION FACTOR
C
C COMPUTE Z TERMS
33    CONTINUE
      Z3=12.0*W(ISLICE)/XMU
      Z4=(R*W(ISLICE)/3600.0)**2
      IF (J.GT.1) GO TO 77
C     DETERMINE INLET CONDITIONS
35    CONTINUE

C     INITIAL STATION COMPUTATIONS -

C BALANCING ON TOTAL PRESSURE -

39    NAG=-1
41    SIGC=(B(J)/APLN(ISLICE))**2*Z4/BETTA(J)
      IF (ABS(SIGB-SIGC).LE..00001*SIGC) GO TO 57
C SIGMA NOT CONVERGED
43    IS=IS+1
      SV(IS)=SIGC
      IF (IS.EQ.3) GO TO 135
45    B(J)=T1-C7*SIGC
      IF (B(J).LT.50.0) GO TO 159
C TEMP OK
47    SIGB=SIGC
      BETTA(J)=BETA1*(B(J)/T1)**C1
      GO TO 41

C SIGMA CONVERGED
57    B(J)=T1-C7*SIGC
      AMC(1)=SQRT(SIGC/B(J))*C5
      IF (B(J).LE.0.0) GO TO 159
      KTRBZ=0
63    BETTA(J)=BETA1/(1.0+C6*AMC(1)**2)**C1
      IF (BETTA(J).LE.0.) GO TO 159
65    B(1)=T1/(1.0+C6*AMC(1)**2)
```

```
      SIGMA(1)=(B(J)/APLN(ISLICE))**2*Z4/BETTA(J)
      SIGC=SQRT(SIGMA(1)/B(1))*C5
      IF (ABS(SIGC-AMC(1)).LE..01) GO TO 71
   67 AMC(1)=SIGC
   69 KTRBZ=KTRBZ+1
      IF (KTRBZ.LE.20) GO TO 63
   71 TT1(1)=TIN(ISLICE)

C    CHANNEL PRESSURE DROP -

   73 NAG = 1
      F1(1)=REY
      SIGMA(2)=SIGMA(1)*.95
      J=2
      IS=0
   77 CONTINUE
      AJ=J-1
      DR2=DR*(2.0*(RIN(ISLICE)+AJ*DR)-DR)
   79 TT1(J)=TT1(J-1)+C3*DR2
      KSIG=0
      AZ=DPLN(ISLICE)/APLN(ISLICE)
      REY=Z3*AZ
      PP = SQRT(BETTA(1))
      SIGMA(1)=SQRT(SIGMA(1))
      BETTA(1)= PP
      TP = B(1)
      G2=(W(ISLICE)/3600.0)**2*R/32.174
      G1= G2/CP/778.161
      D1=(WS*3.1415927/30.0)**2*DR/DX/32.174
      D2= D1/778.161/144.0
      RRP = RIN(ISLICE)
      DO 97  J=2,NSTNS
      AZ=DPLN(ISLICE)/APLN(ISLICE)
      REY= Z3*AZ
      F1(J)= .079*REY**(-.25)
      IF (REY.LT.2300.) F1(J) = 16.0/REY
      PTEMP=BETTA(J-1)
      TTEMP=B(J-1)
      RTEMP = RRP
      MACH1=1
      XNN=2.0
      DD=DPLN(ISLICE)
      AA=APLN(ISLICE)
      GO TO 85
   81 MACH1=XNN
      WRITE(6,83) ISLICE,J
   83 FORMAT(5X,9H******   ,50HDECREASED INCREMENT  DERIVATIVE CHANGING
     1 TOO FAST,3X,'BRANCH NO. ',I2,', STATION NO. ',I2/)
      XNN=XNN*2.0
      DX=DX/2.0
      PTEMP=BETTA(J-1)
      TTEMP=B(J-1)
      PP = PTEMP
      TP = TTEMP
      RRP =RTEMP
      DD= DPLN(ISLICE)
      AA= APLN(ISLICE)
   85 DO 91 L=1,4
      V1=G1/PP/AA**2/(1.0+G1*R*TP/(PP*AA)**2)
      TERM1=TP*G2/(PP*AA)**2
```

```
      TERM2=G1*CP*778.161*R*TP*TP*V1/(PP*AA)**2/PP                    NPLENMP 3181
      TESTMA=1.0-TERM1+TERM2                                          NPLENMP 3182
      IF (TESTMA.LE.0.0) GO TO 159                                    NPLENMP 3183
      XK(L)= FUNP(F1(J))                                              NPLENMP 3184
      IF (L.EQ.1) GO TO 89                                            NPLENMP 3185
      DO 87 LL=2,L                                                    NPLENMP 3186
      XTEST=ABS((XK(L)-XK(LL-1))/PP)                                  NPLENMP 3187
      IF (XTEST.GT.DIFTOL) GO TO 81                                   NPLENMP 3188
87    CONTINUE                                                        NPLENMP 3189
89    XL(L)=FUNT(XK(L))                                               NPLENMP 3190
      IF (L.EQ.4) GO TO 93                                            NPLENMP 3191
      PP =PTEMP+XK(L)/2.0                                             NPLENMP 3192
      TP =TTEMP+XL(L)/2.0                                             NPLENMP 3193
      IF (L.EQ.2)  GO TO 91                                           NPLENMP 3194
      RRP=RRP+DR/XNN                                                  NPLENMP 3195
      IF (L.NE.3)  GO TO 91                                           NPLENMP 3196
      PP =PTEMP+XK(L)                                                 NPLENMP 3197
      TP =TTEMP+XL(L)                                                 NPLENMP 3198
91    CONTINUE                                                        NPLENMP 3199
93    PP =PTEMP+XK(L)                                                 NPLENMP 3200
      TP =TTEMP+XL(L)                                                 NPLENMP 3201
      IF (PP.LE.0.0.OR.TP.LE.0.0) GO TO 159                           NPLENMP 3202
      V1=G1/PP/AA**2/(1.0+G1*R*TP/(PP*AA)**2)                         NPLENMP 3203
      TERM1=TP*G2/(PP*AA)**2                                          NPLENMP 3204
      TERM2=G1*CP*778.161*R*TP*TP*V1/(PP*AA)**2/PP                    NPLENMP 3205
      TESTMA=1.0-TERM1+TERM2                                          NPLENMP 3206
      IF (TESTMA.LE.0.0) GO TO 159                                    NPLENMP 3207
      BETTA(J)=PTEMP+(XK(1)+2.0*(XK(2)+XK(3))+XK(4))/6.0              NPLENMP 3208
      B(J)=TTEMP+(XL(1)+2.0*(XL(2)+XL(3))+XL(4))/6.0                  NPLENMP 3209
      PP = BETTA(J)                                                   NPLENMP 3210
      IF (B(J).LE.0.0.OR.BETTA(J).LE.0.0) GO TO 159                   NPLENMP 3211
      TP =B(J)                                                        NPLENMP 3212
      IF (MACH1.EQ.1) GO TO 95                                        NPLENMP 3213
      MACH1=MACH1-1                                                   NPLENMP 3214
      PTEMP = PP                                                      NPLENMP 3215
      TTEMP = TP                                                      NPLENMP 3216
      GO TO 85                                                        NPLENMP 3217
95    XNN=2.0                                                         NPLENMP 3218
      DX=DXTEMP                                                       NPLENMP 3219
      SIGMA(J)=B(J)/APLN(ISLICE)/BETTA(J)*SQRT(Z4)                    NPLENMP 3220
      AMC(J)=SIGMA(J)*SQRT(1.0/B(J))*C5                               NPLENMP 3221
      IF (AMC(J).GE.1.0) GO TO 159                                    NPLENMP 3222
      F1(J)=REY                                                       NPLENMP 3223
      TT1(J)=B(J)*(1.0+C6*AMC(J)**2)-460.0                            NPLENMP 3224
97    CONTINUE                                                        NPLENMP 3225
C     ALL STATIONS COMPUTED                                           NPLENMP 3226
99    SIGC = 1.0                                                      NPLENMP 3227
      AMC(NSTNS)=AMC(NSTNS)/SIGC                                      NPLENMP 3228
      IF (AMC(NSTNS).GT.1.0)GO TO 159                                 NPLENMP 3229
      BETTA(NSTNS)=BETTA(NSTNS)*SIGC**2                               NPLENMP 3230
C    RESTART CHOKED BRANCH IF M.LT. .8                                NPLENMP 3231
      IF (CH(ISLICE).EQ.0..OR.CH(ISLICE).EQ.1.) GO TO 113             NPLENMP 3232
      IF (ACH.EQ.(-1.0)) GO TO 113                                    NPLENMP 3233
      ACH=-1.                                                         NPLENMP 3234
      AB=0.                                                           NPLENMP 3235
      DO 109 J=1,NSTNS                                                NPLENMP 3236
      IF (AMC(J).GT..8) GO TO 113                                     NPLENMP 3237
      IF (AMC(J).LT.AB) GO TO 109                                     NPLENMP 3238
      AB=AMC(J)                                                       NPLENMP 3239
109   CONTINUE                                                        NPLENMP 3240
```

```
      CH(ISLICE)=0.0                                                    NPLENMP 3241
      AJ=(GAM+1.)/2.                                                    NPLENMP 3242
      CX=-(GAM+1.00)/(GAM-1.0)/2.0                                      NPLENMP 3243
      AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)                NPLENMP 3244
      WCHOKE=W(ISLICE)                                                  NPLENMP 3245
      W(ISLICE)=AZ*W(ISLICE)                                            NPLENMP 3246
      WRITE(6,111) ISLICE,W(ISLICE),WCHOKE                              NPLENMP 3247
111   FORMAT(8X,6H******,23H RESTART CHOKED BRANCH ,I5,24H FLOW RATE INCNPLENMP 3248
     1REASED TO ,F10.4,6H FROM ,F10.4,7H ******)                        NPLENMP 3249
      GO TO 177                                                         NPLENMP 3250
113   BETA1=PIN(ISLICE)**2                                              NPLENMP 3251
C              CALCULATE THE CHOKING FLOW RATE                          NPLENMP 3252
      AB=0.0                                                            NPLENMP 3253
      DO 115 J=1,NSTNS                                                  NPLENMP 3254
      IF (AMC(J).LT.AB) GO TO 115                                       NPLENMP 3255
      AB=AMC(J)                                                         NPLENMP 3256
115   CONTINUE                                                          NPLENMP 3257
      AJ=(GAM+1.0)/2.0                                                  NPLENMP 3258
      CX=-(GAM+1.0)/(GAM-1.0)/2.0                                       NPLENMP 3259
      AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)                NPLENMP 3260
C       COMPUTE  RESISTANCE  EQUATION  FOR  BALANCE                     NPLENMP 3261
      PT1=(1.0+C6*AMC(1)**2)**(C1/2.0)*BETTA(1)                         NPLENMP 3262
      IF (C3.NE.0.0) GO TO 117                                          NPLENMP 3263
      GO TO 119                                                         NPLENMP 3264
117   DR2=ROUT(ISLICE)**2-RIN(ISLICE)**2                                NPLENMP 3265
119   PEXTT=PT1*(1.0+C3/T1*DR2)**(C1/2.0)                               NPLENMP 3266
121   CONTINUE                                                          NPLENMP 3267
      Z1(ISLICE)=(PEXTT**2-BETTA(NSTNS)**2)/W(ISLICE)**2                NPLENMP 3268
      IF (Z1(ISLICE).GT.0.0) GO TO 129                                  NPLENMP 3269
      WRITE(6,125) ISLICE,Z1(ISLICE)                                    NPLENMP 3270
125   FORMAT(///5X,'PASSAGE ',I3,5X,'HAS NEGATIVE OR NO RESISTANCE'     NPLENMP 3271
     Z       ,F12.4//)                                                  NPLENMP 3272
      Z1(ISLICE)=ZED                                                    NPLENMP 3273
129   CONTINUE                                                          NPLENMP 3274
      PP = BETTA(NSTNS)*(1.+C6*AMC(NSTNS)**2)**(C1/2.0)                 NPLENMP 3275
      PTEXIT = PP                                                       NPLENMP 3276
      DIFTOL=0.005                                                      NPLENMP 3277
      KTR1=0                                                            NPLENMP 3278
C                                                                       NPLENMP 3279
C   COMPUTE AVERAGE STATIC PRESSURE AND STATIC TEMPERATURE              NPLENMP 3280
C                                                                       NPLENMP 3281
      PXX = 0.0                                                         NPLENMP 3282
      TXX = 0.0                                                         NPLENMP 3283
      DO 134 I = 1,NSTNS                                                NPLENMP 3284
      TXX = TXX + B(I)                                                  NPLENMP 3285
134   PXX = PXX + BETTA(I)                                              NPLENMP 3286
      TXX = TXX/XXN - 460.                                              NPLENMP 3287
      PXX = PXX/XXN                                                     NPLENMP 3288
      TTEXIT = TT1(NSTNS)                                               NPLENMP 3289
      RETURN                                                            NPLENMP 3290
C                                                                       NPLENMP 3291
CCOMPUTE ACCELERATION                                                   NPLENMP 3292
C                                                                       NPLENMP 3293
135   D=SV(2)-SV(1)                                                     NPLENMP 3294
137   D=(SV(3)-SV(2))/D                                                 NPLENMP 3295
      E=ABS(D)-1.0                                                      NPLENMP 3296
      IF (ABS(D).GT..6) GO TO 143                                       NPLENMP 3297
139   E=D/(D-1.0)                                                       NPLENMP 3298
141   SIGB=E*SV(2)+(1.0-E)*SV(3)                                        NPLENMP 3299
143   IS=0                                                              NPLENMP 3300
```

```
 145    SIGC=SIGB                                                        NPLENMP 3301
        IF (SIGC.LE.0.) GO TO 159                                        NPLENMP 3302
 147    KSIG=KSIG+1                                                      NPLENMP 3303
        IF (KSIG.LT.50) GO TO 155                                        NPLENMP 3304
 149    WRITE (6,151) ISLICE,J                                           NPLENMP 3305
 151    FORMAT(7X,28HPROGRAM IS LOOPING IN BRANCH,I6,9H  STATION,I4//)   NPLENMP 3306
        KSIG=0                                                           NPLENMP 3307
        NCC=NCC+1                                                        NPLENMP 3308
        IF (NCC.LT.4)GO TO 159                                           NPLENMP 3309
 153    ISLICE=220                                                       NPLENMP 3310
        GO TO 129                                                        NPLENMP 3311
 155    CONTINUE                                                         NPLENMP 3312
        IF (NAG.LT.0) GO TO 45                                           NPLENMP 3313
 157    STOP                                                             NPLENMP 3314
C                                                                        NPLENMP 3315
CCHOKING ADJUSTMENT                                                      NPLENMP 3316
C                                                                        NPLENMP 3317
 159    WCHOKE=W(ISLICE)                                                 NPLENMP 3318
        DIFTOL=0.050                                                     NPLENMP 3319
 161    IF (CH(ISLICE).EQ.0.) GO TO 165                                  NPLENMP 3320
 163    W(ISLICE)=.98*W(ISLICE)                                          NPLENMP 3321
        IF (KTR1.GT.20) W(ISLICE)=W(ISLICE)*.98**(KTR1-20)               NPLENMP 3322
        GO TO 171                                                        NPLENMP 3323
 165    AB=0.                                                            NPLENMP 3324
        IF (J.EQ.1) GO TO 169                                            NPLENMP 3325
        DO 167 IJK=1,J                                                   NPLENMP 3326
        IF (AMC(IJK).GT.1.0) AMC(IJK)=1.0                                NPLENMP 3327
        IF (AMC(IJK).LT.AB) GO TO 167                                    NPLENMP 3328
        AB=AMC(IJK)                                                      NPLENMP 3329
 167    CONTINUE                                                         NPLENMP 3330
        AJ=(GAM+1.)/2.                                                   NPLENMP 3331
        CX=-(GAM+1.00)/(GAM-1.0)/2.0                                     NPLENMP 3332
        AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)               NPLENMP 3333
        W(ISLICE)=AZ*W(ISLICE)                                           NPLENMP 3334
        GO TO 171                                                        NPLENMP 3335
 169    CONTINUE                                                         NPLENMP 3336
        W(ISLICE)=1600.0*APLN(ISLICE)*SQRT((32.17*BETA1*GAM)/(R*T1))     NPLENMP 3337
 171    CH(ISLICE)=1.0                                                   NPLENMP 3338
 173    WRITE(6,175) ISLICE,J,W(ISLICE),WCHOKE                           NPLENMP 3339
 175    FORMAT(3X,12H*** PASSAGE ,I5,23H HAS CHOKED AT STATION ,I5,      NPLENMP 3340
       134H AND THE FLOW HAS BEEN REDUCED TO ,F10.4,6H FROM ,F10.4,4H ***)NPLENMP 3341
 177    IS=0                                                             NPLENMP 3342
        BETTA(1)=PIN(ISLICE)**2                                          NPLENMP 3343
        B(1)=T1                                                          NPLENMP 3344
        DX=DXTEMP                                                        NPLENMP 3345
        J=1                                                              NPLENMP 3346
        SIGB=0.0                                                         NPLENMP 3347
        KTR1=KTR1+1                                                      NPLENMP 3348
        IF (KTR1.GE.50) GO TO 181                                        NPLENMP 3349
 179    GO TO 17                                                         NPLENMP 3350
 181    WRITE (6,183) ISLICE                                             NPLENMP 3351
 183    FORMAT(/2X,16H**FLOW IN BRANCH,I6,                               NPLENMP 3352
       Z        ' HAS BEEN REDUCED 50 TIMES BECAUSE OF CHOKING')         NPLENMP 3353
        GO TO 153                                                        NPLENMP 3354
        END                                                              NPLENMP 3355


C----SOURCE.NPLOTIT                                                      NPLOTIT 3356
        SUBROUTINE PLOTMF(ALPH2)                                         NPLOTIT 3357
C                                                                        NPLOTIT 3358
C     SOURCE.NPLOTIT                                                     NPLOTIT 3359
C                                                                        NPLOTIT 3360
```

```
      COMMON /PRPS/ CPO,       GAMO,       DP(80),     SP(80),  RE(80),
     Z              CPC(80),   GAMC(80),   DUMR1(80),  DUMR2(80)
C
      COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),
     Z               IPLOT,  MD1,  MD2,  MD3, IADJIN, IWRITE
C
      COMMON /TCO/ ADUMP,      ETA,        CD,          CP,
     Z             GAM,        PIM,        R,           SPAN,       TOG,
     Z             WDUMP,      WIM,        AKC(15,80),  AKW(15,80),
     Z             A(400),     AJET(80),   AM2(80),     CNUM(80),
     Z             DH(80),     DHF(80),    DHJ(80),
     Z             DLX(400),   FF(80),     HC(80),      HG(80),
     Z             P(2,15,80),PEXIT(15),   PUMP(80),    QG(80),
     Z             QSNK(80),   RR(80),     S(15),       T(2,15,400),
     Z             TG(80),     TAU(400),   WFC(80),
     Z             WJ(15,80),  WCROS(2,15,80),          XN(80),
     Z             ICOR,       IFILM,      IHUB,        ITIP,
     Z             ISBLOK,     ISLICE,     NBLKSZ,      NSLICE,
     Z             NFWD,       NSTA,       IHC(80)
C
      COMMON /TRNSNT/ RHOC,       RHOM,       SPHTC,       SPHTM,
     Z                DLTYME,     TYME,       TEPS,        TYMMAX
C
      COMMON /UNITS/ CINCH(2), CHTC(2),  CHFLX(2), CPRSR(2), CMSFL(2),
     Z               CTMPF(2), CTCON(2), CDEN(2),  CSPHT(2), CGASC(2),
     Z               CVISC(2), CRHOVG(2),IUNITS
C
      DIMENSION Y(320),XLABL(29),YLABL(7),TLABL1(21),TLABL2(9)
      DIMENSION XLABL2(15),VARIB(15),YPLABL(10), ALPH2(4), ALABL(7)
      DIMENSION XS(80), XP(80), TSO(500),TSM(500), TPO(500), TPM(500)
      DIMENSION  YLABL2(11)
      DIMENSION PLEGN(5), SLEGN(5), SYMBL(10), YLBL(20), XLBL(20)
      DIMENSION  YTEM(80), ISYM(5), PLTYME(2)
      LOGICAL*1 IXAX/.TRUE./, IYAX/.FALSE./
      INTEGER*2 NPTS
      DIMENSION RTNARR(2),VARS(12),IVARS(12)
      DATA PLEGN/'A)  ','PRES','SURE',' SID','E   '/
      DATA SLEGN/'B)  ','SUCT','ION ',' SID','E   '/
      DATA XLABL2/15*'    '/
      DATA SYMBOL/'0'/
      DATA XLABL/'MID-','WALL',' X/L',' PRO','M ST','ATIO','N NO','. 1,'
     Z          ,' L =','    ','    ',' INC','HES ','    ','    ','    '
     Z          ,'    ',' SID','E; C','HNL ','    ','; PL','ENUM',' PRE'
     Z          ,'SS.=','    ','    ',' PSI','A***'/
      DATA YLABL/'----','TEMP','ERAT','URE,',' DEG','. F.','----'/
      DATA YPLABL/'  CO','OLAN','T CH','ANNE','L  T','OTAL',' PRE'
     Z           ,'SSUR','E,  ','PSIA'/
      DATA YLABL2/7*'    ','TEMP','ERAT','URE,',' F  '/
      DATA TLABL1/21*'    '/
      DATA TLABL2/9*'    '/
      DATA VARIB/'PRES','SURE',' SUC','TION','SURF','ACE ','MID-',
     Z    'WALL',' CM ','    ',' KPA','****',' K  ','  KP','A   '/
      DATA ALABL/2*'    ','----',2*'    ','----','    '/
      DATA SYMBL/'1','2','3','4','5','6','7','8','9','0'/
      DATA ISYM/62,119,118,70,65/
      DATA PLTYME/2*'    '/
      ATYME = TYME
      IF (ATYME.LT.0.0) ATYME = 0.0
      CALL NUMBER(4,ATYME,8,2,PLTYME)
      ALABL(6) = PLTYME(1)
```

```
      ALABL(7) = PLTYME(2)
C
C  MD2 > 0 INDICATES JOB IS COMPLETE.--NOW DO SUMMARY PLOTS.
C
10    CONTINUE
      IF (MD2.GT.0) GO TO 80
20    NSTAPS = NSTA/2 + 1
      NLBLS = NSTAPS/5
C
C  SET UP TIME AND DATE LABEL FOR PLOT IDENTIFICATION
C
      ALABL(1) = ALPH2(3)
      ALABL(2) = ALPH2(4)
      ALABL(3) = ALPH2(1)
      ALABL(4) = ALPH2(2)
C
C   SET UP TITLE
C
      DO 45 I = 1,30
      IF (I.LE.21) TLABL1(I) = TITLE(I)
      IF (I.GT.21) TLABL2(I-21) = TITLE(I)
45    CONTINUE
C
C PRESSURE SIDE
C
46    IF (MD3.GT.1) GO TO 55
47    XP(1) = 0.0
      IX = 1
      DO 50 I = 3,NSTA,2
C
C  NMM IS THE MIDMETAL NODE NUMBER  (L)
C
      NMM = 5*I - 2
      IX = IX + 1
50    XP(IX) = XP(IX-1) + DLX(NMM)/CINCH(IUNITS)
      XPL = XP(NSTAPS)
      DO 51 I = 2,NSTAPS
51    XP(I) = XP(I)/XPL
55    CONTINUE
      IY = 0
      ITP = NSTAPS*(ISLICE-1)
      DO 60 I = 1,NSTA,2
      IY = IY + 1
      NOS = 5*I - 4
      Y(IY) = T(2,ISLICE,NOS)/CTMPF(IUNITS)
      IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.
      ITP = ITP + 1
      TPO(ITP) = Y(IY)
      IYP = IY + NSTAPS
      Y(IYP) = T(2,ISLICE,NOS+1)/CTMPF(IUNITS)
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
      IYP = IY + 2*NSTAPS
      Y(IYP) = T(2,ISLICE,NOS+2)/CTMPF(IUNITS)
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
      TPM(ITP) = Y(IYP)
      IYP = IY + 3*NSTAPS
      Y(IYP) = T(2,ISLICE,NOS+3)/CTMPF(IUNITS)
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
      IYP = IY + 4*NSTAPS
      NCOOL = NOS + 4
```

```
      Y(IYP) = T(2,ISLICE,NCOOL)/CTMPF(IUNITS)
60    IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
      XLABL(16) = VARIB(1)
      XLABL(17) = VARIB(2)
      IF (IUNITS.EQ.2) GO TO 601
      XLABL(12) = VARIB(9)
      XLABL(13) = VARIB(10)
      XLABL(28) = VARIB(11)
      XLABL(29) = VARIB(12)
      YLABL(5) = VARIB(13)
      YLABL(6) = VARIB(10)
      YPLABL(10) = VARIB(11)
      YLABL2(11) = VARIB(13)
601   CONTINUE
      DO 611 I = 1,15
611   XLABL2(I) = XLABL(I+15)
C
      IF (IPLOT.EQ.3) GO TO 63
C
      CPIM = PIM/CPRSR(IUNITS)
      CALL NUMBER(1,ISLICE,4,0,XLABL2(6))
      CALL NUMBER(4,CPIM,8,1,XLABL2(11))
      CALL NUMBER(4,XPL,8,4,XLABL(10))
      CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)
      CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)
      CALL CHARS(60,XLABL,0.0,1.5,.25,12)
      CALL CHARS(56,XLABL2,0.0,1.5,.05,12)
      CALL CHARS(28,YLABL,90.,.25,3.3,12)
      MD3 = MD3 + 1
      CALL NUMBER(1,MD3,4,0,ALABL(5))
      CALL CHARS(28,ALABL,0.0,6.2,9.3,12)
C
C--- TITLES ARE DONE, NOW SET UP AXES FOR TEMPERATURE PLOTS
C
      NPTS = NSTAPS
      CALL SCALE(IXAX,NPTS,XP)
      NPTS = 5*NSTAPS
      CALL SCLBAK(IYAX,NPTS,Y,RTNARR)
      CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)
      VARS(1) = 7.0
      VARS(2) = 9.0
      VARS(3) = 0.0
      VARS(4) = 0.0
      VARS(5) = 1.0
      VARS(6) = .5
      VARS(7) = 1.0
      CALL XAXIS(.8,.6,VARS)
      VARS(2) = 8.9
      VARS(3) = 90.
      VARS(4) = YMIN
      VARS(5) = YMAX
      CALL YAXIS(.8,.6,VARS)
C
C--- AXES ARE SET. NOW PLOT THE FIVE TEMPERATURE CURVES, USING
C     DIFFERENT SYMBOLS FOR EACH.
C
      IVARS(1) = 4
      IVARS(2) = NSTAPS
      IVARS(3) = 2
      DO 603 I = 1,5
```

```
      IVARS(4) = ISYM(I)
      IYST = 1 + (I-1)*NSTAPS
      IYEN = I*NSTAPS
      III = 0
      DO 602 II = IYST,IYEN
      III = III + 1
602   YTEM(III) = Y(II)
      CALL GPLOT(XP,YTEM,IVARS)
603   CONTINUE
      CALL DISPLA(1)
C
C
C  PRESSURE SIDE COOLANT PRESSURE DISTRIBUTION
C
      IY = 0
      DO 61 I = 1,NSTA,2
      IY = IY + 1
61    Y(IY) = P(2,ISLICE,I)*
     Z (1.+(GAMC(I)-1.)*AM2(I)/2.)**(GAMC(I)/(GAMC(I)-1.))/CPRSR(IUNITS)
      CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)
      CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)
      CALL CHARS(60,XLABL,0.0,1.5,.25,12)
      CALL CHARS(56,XLABL2,0.0,1.5,.05,12)
      CALL CHARS(40,YPLABL,90.,.25,2.8,12)
      MD3 = MD3 + 1
      CALL NUMBER(1,MD3,4,0,ALABL(5))
      CALL CHARS(28,ALABL,0.0,6.2,9.3,12)
      NPTS = NSTAPS
      CALL SCALE(IXAX,NPTS,XP)
      CALL SCLBAK(IYAX,NPTS,Y,RTNARR)
      CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)
      VARS(1) = 7.0
      VARS(2) = 9.0
      VARS(3) = 0.0
      VARS(4) = 0.0
      VARS(5) = 1.0
      VARS(6) = .5
      VARS(7) = 1.0
      CALL XAXIS(.8,.6,VARS)
      VARS(2) = 8.9
      VARS(3) = 90.
      VARS(4) = YMIN
      VARS(5) = YMAX
      CALL YAXIS(.8,.6,VARS)
C
C--- AXES ARE SET, NOW PLOT THE PRESSURE
C
      IVARS(1) = 4
      IVARS(2) = NSTAPS
      IVARS(3) = 2
      IVARS(4) = 65
      CALL GPLOT(XP,Y,IVARS)
      CALL DISPLA(1)
C
C  SUCTION SIDE
C
      IF (MD3.GT.2) GO TO 69
63    XS(1) = 0.0
      IX = 1
      DO 65 I = 2,NSTA,2
```

```
      NMM = 5*I - 2                                                     NPLOTIT 3601
      IX = IX + 1                                                       NPLOTIT 3602
65    XS(IX) = XS(IX-1) + DLX(NMM)/CINCH(IUNITS)                        NPLOTIT 3603
      XSL = XS(NSTAPS)                                                  NPLOTIT 3604
      DO 66 I = 2,NSTAPS                                                NPLOTIT 3605
66    XS(I) = XS(I)/XSL                                                 NPLOTIT 3606
69    CONTINUE                                                          NPLOTIT 3607
C                                                                       NPLOTIT 3608
      IY = 1                                                            NPLOTIT 3609
      ITS = NSTAPS*(ISLICE-1) + 1                                       NPLOTIT 3610
      Y(IY) = T(2,ISLICE,1)/CTMPF(IUNITS)                               NPLOTIT 3611
      IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.                             NPLOTIT 3612
      TSO(ITS) = Y(IY)                                                  NPLOTIT 3613
      IYP = IY + NSTAPS                                                 NPLOTIT 3614
      Y(IYP) = T(2,ISLICE,2)/CTMPF(IUNITS)                              NPLOTIT 3615
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.                           NPLOTIT 3616
      IYP = IYP + NSTAPS                                                NPLOTIT 3617
      Y(IYP) = T(2,ISLICE,3)/CTMPF(IUNITS)                              NPLOTIT 3618
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.                           NPLOTIT 3619
      TSM(ITS) = Y(IYP)                                                 NPLOTIT 3620
      IYP = IYP + NSTAPS                                                NPLOTIT 3621
      Y(IYP) = T(2,ISLICE,4)/CTMPF(IUNITS)                              NPLOTIT 3622
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.                           NPLOTIT 3623
      IYP = IYP + NSTAPS                                                NPLOTIT 3624
      Y(IYP) = T(2,ISLICE,5)/CTMPF(IUNITS)                              NPLOTIT 3625
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.                           NPLOTIT 3626
      DO 70 I = 2,NSTA,2                                                NPLOTIT 3627
      IY = IY + 1                                                       NPLOTIT 3628
      NOS = 5*I - 4                                                     NPLOTIT 3629
      Y(IY) = T(2,ISLICE,NOS)/CTMPF(IUNITS)                             NPLOTIT 3630
      IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.                             NPLOTIT 3631
      ITS = ITS + 1                                                     NPLOTIT 3632
      TSO(ITS) = Y(IY)                                                  NPLOTIT 3633
      IYP = IY + NSTAPS                                                 NPLOTIT 3634
      Y(IYP) = T(2,ISLICE,NOS+1)/CTMPF(IUNITS)                          NPLOTIT 3635
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.                           NPLOTIT 3636
      IYP = IYP + NSTAPS                                                NPLOTIT 3637
      Y(IYP) = T(2,ISLICE,NOS+2)/CTMPF(IUNITS)                          NPLOTIT 3638
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.                           NPLOTIT 3639
      TSM(ITS) = Y(IYP)                                                 NPLOTIT 3640
      IYP = IYP + NSTAPS                                                NPLOTIT 3641
      Y(IYP) = T(2,ISLICE,NOS+3)/CTMPF(IUNITS)                          NPLOTIT 3642
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.                           NPLOTIT 3643
      IYP = IYP + NSTAPS                                                NPLOTIT 3644
      NCOOL = NOS + 4                                                   NPLOTIT 3645
      Y(IYP) = T(2,ISLICE,NCOOL)/CTMPF(IUNITS)                          NPLOTIT 3646
70    IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.                           NPLOTIT 3647
      XLABL2(1) = VARIB(3)                                              NPLOTIT 3648
      XLABL2(2) = VARIB(4)                                              NPLOTIT 3649
C                                                                       NPLOTIT 3650
      IF (IPLOT.EQ.3) GO TO 80                                          NPLOTIT 3651
C                                                                       NPLOTIT 3652
      CALL NUMBER(1,ISLICE,4,0,XLABL2(6))                               NPLOTIT 3653
      CALL NUMBER(4,CPIM,8,1,XLABL2(11))                                NPLOTIT 3654
      CALL NUMBER(4,XSL,8,4,XLABL(10))                                  NPLOTIT 3655
      IX = 2 + NFWD/2                                                   NPLOTIT 3656
      CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)                            NPLOTIT 3657
      CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)                            NPLOTIT 3658
      CALL CHARS(60,XLABL,0.0,1.5,.25,12)                               NPLOTIT 3659
      CALL CHARS(56,XLABL2,0.0,1.5,.05,12)                              NPLOTIT 3660
```

```
      CALL CHARS(28,YLABL,90.,.25,3.3,12)
      MD3 = MD3 + 1
      CALL NUMBER(1,MD3,4,0,ALABL(5))
      CALL CHARS(28,ALABL,0.0,6.2,9.3,12)
C
C--- TITLES ARE DONE, NOW SET UP AXES FOR TEMPERATURE PLOTS
C
      NPTS = NSTAPS
      CALL SCALE(IXAX,NPTS,XS)
      NPTS = 5*NSTAPS
      CALL SCLBAK(IYAX,NPTS,Y,RTNARR)
      CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)
      VARS(1) = 7.0
      VARS(2) = 9.0
      VARS(3) = 0.0
      VARS(4) = 0.0
      VARS(5) = 1.0
      VARS(6) = .5
      VARS(7) = 1.0
      CALL XAXIS(.8,.6,VARS)
      VARS(2) = 8.9
      VARS(3) = 90.
      VARS(4) = YMIN
      VARS(5) = YMAX
      CALL YAXIS(.8,.6,VARS)
C
C--- AXES ARE SET. NOW PLOT THE FIVE TEMPERATURE CURVES, USING
C     DIFFERENT SYMBOLS FOR EACH.
C
      IVARS(1) = 4
      IVARS(2) = NSTAPS
      IVARS(3) = 2
      DO 703 I = 1,5
      IVARS(4) = ISYM(I)
      IYST = 1 + (I-1)*NSTAPS
      IYEN = I*NSTAPS
      III = 0
      DO 702 II = IYST,IYEN
      III = III + 1
702   YTEM(III) = Y(II)
      CALL GPLOT(XS,YTEM,IVARS)
703   CONTINUE
      CALL DISPLA(1)
C
C
C  SUCTION SIDE COOLANT PRESSURE DISTRIBUTION
C
      Y(1) = P(2,ISLICE,1)*
     Z (1.+(GAMC(1)-1.)*AM2(1)/2.)**(GAMC(1)/(GAMC(1)-1.))/CPRSR(IUNITS)
      IY = 1
      DO 75 I = 2,NSTA,2
      IY = IY + 1
75    Y(IY) = P(2,ISLICE,I)*(1.+(GAMC(I)-1.)
     Z        *AM2(I)/2.)**(GAMC(I)/(GAMC(I)-1.))/CPRSR(IUNITS)
      CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)
      CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)
      CALL CHARS(60,XLABL,0.0,1.5,.25,12)
      CALL CHARS(56,XLABL2,0.0,1.5,.05,12)
      CALL CHARS(40,YPLABL,90.,.25,2.8,12)
      MD3 = MD3 + 1
```

```
      CALL NUMBER(1,MD3,4,0,ALABL(5))
      CALL CHARS(28,ALABL,0.0,6.2,9.3,12)
      NPTS = NSTAPS
      CALL SCALE(IXAX,NPTS,XS)
      CALL SCLBAK(IYAX,NPTS,Y,RTNARR)
      CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)
      VARS(1) = 7.0
      VARS(2) = 9.0
      VARS(3) = 0.0
      VARS(4) = 0.0
      VARS(5) = 1.0
      VARS(6) = .5
      VARS(7) = 1.0
      CALL XAXIS(.8,.6,VARS)
      VARS(2) = 8.9
      VARS(3) = 90.
      VARS(4) = YMIN
      VARS(5) = YMAX
      CALL YAXIS(.8,.6,VARS)
C
C--- AXES ARE SET, NOW PLOT THE PRESSURE
C
      IVARS(1) = 4
      IVARS(2) = NSTAPS
      IVARS(3) = 2
      IVARS(4) = 65
      CALL GPLOT(XS,Y,IVARS)
      CALL DISPLA(1)
      GO TO 150
C
C  THE FOLLOWING SECTION PUTS OUT PLOTS CONTAINING TEMPERATURES FROM
C    ALL SLICES ON ONE FRAME
C
80    CONTINUE
      IF (ISLICE.LT.NSLICE) GO TO 150
C
C   THE FOLLOWING PUTS TWO PLOTS ON ONE FRAME OF FILM
C
C  FIRST PLOT THE OUTSIDE SURFACE TEMPERATURES FOP EACH
C    SLICE ON THE SAME PLOT
C
      NPTS = NSTAPS*NSLICE
      CALL SCLBAK(IYAX,NPTS,TPC,RTNARR)
      TMAXP = RTNARR(2)
      TMINP = RTNARR(1)
      CALL SCLBAK(IYAX,NPTS,TSO,RTNARR)
      TMAXS = RTNARR(2)
      TMINS = PTNARR(1)
      IF (TMAXS.GT.TMAXP) TMAXP = TMAXS
      IF (TMINS.LT.TMINP) TMINP = TMINS
      NINTRV = (TMAXP-TMINP)/100. + 2
      CALL GINTVL(TMINP,TMAXP,NINTRV,0,ATMINP,ATMAXP)
      AINTRV = NINTRV
      CALL CHAPS(84,TLABL1,0.0,0.15,9.85,12)
      CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)
      YLABL2(5) = VARIB(5)
      YLABL2(6) = VARIB(6)
      CALL CHARS(44,YLABL2,90.,.25,1.6,12)
      VARS(1) = 8.
      VARS(2) = 8.5
```

```
      VARS(3)  =  0.
      VARS(4)  =  0.0
      VARS(5)  =  1.0
      VARS(6)  =  .5
      VARS(7)  =  1.0
      VARS(8)  =  0.0
      CALL XAXIS(1.2,5.5,VARS)
      MD3 = MD3 + 1
      CALL NUMBER(1,MD3,4,0,ALABL(5))
      CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
      CALL CHARS(20,PLEGN,0.0,6.0,9.5,12)
      VARS(1)  =  7.
      VARS(2)  =  3.8
      VARS(3)  =  90.
      VARS(4)  =  ATMINP
      VARS(5)  =  ATMAXP
      VARS(6)  =  AINTRV
      VARS(7)  =  1.
      CALL YAXIS(1.2,5.5,VARS)
      DO 100 I = 1,NSLICE
      JST = NSTAPS*(I-1)
      DO 95 J = 1,NSTAPS
95    Y(J)  =  TPO(JST+J)
      SYMBOL = SYMBL(I)
      KS = 0
C
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO
C     IDENTIFY THE CURVES.
      KSTART = I + 1
      DO 98 K = KSTART,NSTAPS,10
      KS = KS + 1
      XLBL(KS)  =  XP(K)
98    YLBL(KS)  =  Y(K)
      IVARS(1)  =  6
      IVARS(2)  =  NLBLS
      IVARS(3)  =  3
      IVARS(4)  =  240+I
      IVARS(5)  =  1
      IVARS(6)  =  8
      CALL GPLOT(XLBL,YLBL,IVARS)
      IVARS(1)  =  3
      IVARS(2)  =  NSTAPS
      IVARS(3)  =  0
100   CALL GPLOT(XP,Y,IVARS)
C
C
      VARS(1)  =  7.
      VARS(2)  =  8.5
      VARS(3)  =  0.
      VARS(4)  =  0.0
      VARS(5)  =  1.0
      VARS(6)  =  .5
      VARS(7)  =  1.0
      CALL XAXIS(1.2,.5,VARS)
      CALL CHARS(42,XLABL,0.0,3.0,.05,12)
      VARS(1)  =  7.
      VARS(2)  =  3.8
      VARS(3)  =  90.
      VARS(4)  =  ATMINP
      VARS(5)  =  ATMAXP
```

```
      VARS(6) = AINTRV                                                  NPLOTIT 3841
      VARS(7) = 1.                                                      NPLOTIT 3842
      CALL YAXIS(1.2,.5,VARS)                                           NPLOTIT 3843
      CALL CHARS(20,SLEGN,0.0,6.0,4.5,12)                               NPLOTIT 3844
      DO 110 I = 1,NSLICE                                               NPLOTIT 3845
      JST = NSTAPS*(I-1)                                                NPLOTIT 3846
      DO 105 J = 1,NSTAPS                                               NPLOTIT 3847
105   Y(J) = TSO(JST+J)                                                 NPLOTIT 3848
      SYMBOL = SYMBL(I)                                                 NPLOTIT 3849
      KS = 0                                                            NPLOTIT 3850
C                                                                       NPLOTIT 3851
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO                    NPLOTIT 3852
C     IDENTIFY THE CURVES.                                              NPLOTIT 3853
      KSTART = I + 1                                                    NPLOTIT 3854
      DO 108 K = KSTART,NSTAPS,10                                       NPLOTIT 3855
      KS = KS + 1                                                       NPLOTIT 3856
      XLBL(KS) = XS(K)                                                  NPLOTIT 3857
108   YLBL(KS) = Y(K)                                                   NPLOTIT 3858
      IVARS(1) = 6                                                      NPLOTIT 3859
      IVARS(2) = NLBLS                                                  NPLOTIT 3860
      IVARS(3) = 3                                                      NPLOTIT 3861
      IVARS(4) = 240+I                                                  NPLOTIT 3862
      IVARS(5) = 1                                                      NPLOTIT 3863
      IVARS(6) = 8                                                      NPLOTIT 3864
      CALL GPLOT(XLBL,YLBL,IVARS)                                       NPLOTIT 3865
      IVARS(1) = 3                                                      NPLOTIT 3866
      IVARS(2) = NSTAPS                                                 NPLOTIT 3867
      IVARS(3) = 0                                                      NPLOTIT 3868
110   CALL GPLOT(XS,Y,IVARS)                                            NPLOTIT 3869
      CALL DISPLA(1)                                                    NPLOTIT 3870
C                                                                       NPLOTIT 3871
C                                                                       NPLOTIT 3872
C                                                                       NPLOTIT 3873
C  NOW PLOT THE MID-METAL TEMPERATURES FOR EACH SLICE, ALL ON ONE PLOT  NPLOTIT 3874
C                                                                       NPLOTIT 3875
112   CONTINUE                                                          NPLOTIT 3876
      CALL SCLBAK(IYAX,NPTS,TPM,RTNARR)                                 NPLOTIT 3877
      TMAXP = RTNARR(2)                                                 NPLOTIT 3878
      TMINP = RTNARR(1)                                                 NPLOTIT 3879
      CALL SCLBAK(IYAX,NPTS,TSM,RTNARR)                                 NPLOTIT 3880
      TMAXS = RTNARR(2)                                                 NPLOTIT 3881
      TMINS = RTNARR(1)                                                 NPLOTIT 3882
      IF (TMAXS.GT.TMAXP) TMAXP = TMAXS                                 NPLOTIT 3883
      IF (TMINS.LT.TMINP) TMINP = TMINS                                 NPLOTIT 3884
      NINTRV = (TMAXP-TMINP)/100. + 2                                   NPLOTIT 3885
      CALL GINTVL(TMINP,TMAXP,NINTRV,0,ATMINP,ATMAXP)                   NPLOTIT 3886
      AINTRV = NINTRV                                                   NPLOTIT 3887
      CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)                            NPLOTIT 3888
      CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)                            NPLOTIT 3889
      YLABL2(5) = VARIB(7)                                              NPLOTIT 3890
      YLABL2(6) = VARIB(8)                                              NPLOTIT 3891
      CALL CHARS(44,YLABL2,90.,.25,1.6,12)                              NPLOTIT 3892
      VARS(1) = 8.                                                      NPLOTIT 3893
      VARS(2) = 8.5                                                     NPLOTIT 3894
      VARS(3) = 0.                                                      NPLOTIT 3895
      VARS(4) = 0.0                                                     NPLOTIT 3896
      VARS(5) = 1.0                                                     NPLOTIT 3897
      VARS(6) = .5                                                      NPLOTIT 3898
      VARS(7) = 1.0                                                     NPLOTIT 3899
      VARS(8) = 0.0                                                     NPLOTIT 3900
```

```
      CALL XAXIS(1.2,5.5,VARS)
      MD3 = MD3 + 1
      CALL NUMBER(1,MD3,4,0,ALABL(5))
      CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
      CALL CHARS(20,PLEGN,0.0,6.0,9.5,12)
      VARS(1) = 7.
      VARS(2) = 3.8
      VARS(3) = 90.
      VARS(4) = ATMINP
      VARS(5) = ATMAXP
      VARS(6) = AINTRV
      VARS(7) = 1.
      CALL YAXIS(1.2,5.5,VARS)
114   CONTINUE
      DO 120 I = 1,NSLICE
      JST = NSTAPS*(I-1)
      DO 115 J = 1,NSTAPS
115   Y(J) = TPM(JST+J)
      SYMBOL = SYMBL(I)
      KS = 0
C
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO
C     IDENTIFY THE CURVES.
      KSTART = I + 1
      DO 118 K = KSTART,NSTAPS,10
      KS = KS + 1
      XLBL(KS) = XP(K)
118   YLBL(KS) = Y(K)
      IVARS(1) = 6
      IVARS(2) = NLBLS
      IVARS(3) = 3
      IVARS(4) = 240+I
      IVARS(5) = 1
      IVARS(6) = 8
      CALL GPLOT(XLBL,YLBL,IVARS)
      IVARS(1) = 3
      IVARS(2) = NSTAPS
      IVARS(3) = 0
120   CALL GPLOT(XP,Y,IVARS)
C
C
      VARS(1) = 7.
      VARS(2) = 8.5
      VARS(3) = 0.
      VARS(4) = 0.0
      VARS(5) = 1.0
      VARS(6) = .5
      VARS(7) = 1.0
      CALL XAXIS(1.2,.5,VARS)
      CALL CHARS(42,XLABL,0.0,3.5,.05,15)
      VARS(1) = 7.
      VARS(2) = 3.8
      VARS(3) = 90.
      VARS(4) = ATMINP
      VARS(5) = ATMAXP
      VARS(6) = AINTRV
      VARS(7) = 1.
      CALL YAXIS(1.2,.5,VARS)
      CALL CHARS(20,SLEGN,0.0,6.0,4.5,12)
      DO 130 I = 1,NSLICE
```

```
      JST = NSTAPS*(I-1)                                                NPLOTIT 3961
      DO 125 J = 1,NSTAPS                                               NPLOTIT 3962
125   Y(J) = TSM(JST+J)                                                 NPLOTIT 3963
      SYMBOL = SYMBL(I)                                                 NPLOTIT 3964
      KS = 0                                                            NPLOTIT 3965
C                                                                       NPLOTIT 3966
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO                    NPLOTIT 3967
C     IDENTIFY THE CURVES.                                              NPLOTIT 3968
      KSTART = I + 1                                                    NPLOTIT 3969
      DO 128 K = KSTART,NSTAPS,10                                       NPLOTIT 3970
      KS = KS + 1                                                       NPLOTIT 3971
      XLBL(KS) = XS(K)                                                  NPLOTIT 3972
128   YLBL(KS) = Y(K)                                                   NPLOTIT 3973
      IVARS(1) = 6                                                      NPLOTIT 3974
      IVARS(2) = NLBLS                                                  NPLOTIT 3975
      IVARS(3) = 3                                                      NPLOTIT 3976
      IVARS(4) = 240+I                                                  NPLOTIT 3977
      IVARS(5) = 1                                                      NPLOTIT 3978
      IVARS(6) = 8                                                      NPLOTIT 3979
      CALL GPLOT(XLBL,YLBL,IVARS)                                       NPLOTIT 3980
      IVARS(1) = 3                                                      NPLOTIT 3981
      IVARS(2) = NSTAPS                                                 NPLOTIT 3982
      IVARS(3) = 0                                                      NPLOTIT 3983
130   CALL GPLOT(XS,Y,IVARS)                                            NPLOTIT 3984
      CALL DISPLA(1)                                                    NPLOTIT 3985
C                                                                       NPLOTIT 3986
C                                                                       NPLOTIT 3987
150   CONTINUE                                                          NPLOTIT 3988
      RETURN                                                            NPLOTIT 3989
      END                                                               NPLOTIT 3990


C----SOURCE.NPREPAT                                                     NPREPAT 3991
      SUBROUTINE PREP(ICHNL,NTTG)                                       NPREPAT 3992
C                                                                       NPREPAT 3993
C-    SOURCE.NPREPAT----                                                NPREPAT 3994
C                                                                       NPREPAT 3995
      COMMON /BOUND/ BCXS(400),   BCXP(400),   BCHGS(1000), BCHGP(1000),NPREPAT 3996
     Z               BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),NPREPAT 3997
     Z               BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),   NPREPAT 3998
     Z               QHUBIN(400), QHUB(80),    TTIPIN(400), TTIP(80),   NPREPAT 3999
     Z               QTIPIN(400), QTIP(80),    RHOVG(400),  PEX(400),   NPREPAT 4000
     Z               BCTIME(50),  TTIO(50),    PTIO(50),    WPLEN,      NPREPAT 4001
     Z               WSVST(50),   AKCTBL(20),  AKWTBL(20),  NBCS,   NBCP NPREPAT 4002
C                                                                       NPREPAT 4003
      COMMON /FLMCOL/ RHOVGA(80), PG(80),   XFC(80),   FLMEFF(80),      NPREPAT 4004
     Z                XMUC(80),   EMES(80), REFC(80), NFCSUP(80)        NPREPAT 4005
C                                                                       NPREPAT 4006
      COMMON /FRIC/ ALPHA,  BETA,  DELTA,  EPS                          NPREPAT 4007
C                                                                       NPREPAT 4008
      COMMON /PRPS/ CPO,      GAMO,     DP(80),     SP(80),   RE(80),   NPREPAT 4009
     Z              CPC(80),  GAMC(80), DUMR1(80),  DUMR2(80)           NPREPAT 4010
C                                                                       NPREPAT 4011
      COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),              NPREPAT 4012
     Z               IPLOT,  MD1,  MD2,  MD3, IADJIN, IWRITE            NPREPAT 4013
C                                                                       NPREPAT 4014
      COMMON /TCC/ ADUMP,     BTA,      CD,         CP,                 NPREPAT 4015
     Z             GAM,       PIM,      R,          SPAN,        TOG,   NPREPAT 4016
     Z             WDUMP,     WIM,      AKC(15,80), AKW(15,80),         NPREPAT 4017
     Z             A(400),    AJET(80), AM2(80),    CNUM(80),           NPREPAT 4018
     Z             DH(80),    DHF(80),  DHJ(80),                        NPREPAT 4019
     Z             DLX(400),  FF(80),   HC(80),     HG(80),             NPREPAT 4020
```

```
     Z              P(2,15,80),PEXIT(15),  PUMP(80),    QG(80),
     Z              QSNK(80),  RR(80),     S(15),       T(2,15,400),
     Z              TG(80),    TAU(400),   WFC(80),
     Z              WJ(15,80), WCROS(2,15,80),          XN(80),
     Z              ICOR,      IFILM,      IHUB,        ITIP,
     Z              ISBLOK,    ISLICE,     NBLKSZ,      NSLICE,
     Z              NFWD,      NSTA,       IHC(80)
C
      COMMON /TRNSNT/ RHOC,       RHOM,       SPHTC,       SPHTM,
     Z                DLTYME,     TYME,       TEPS,        TYMMAX
C
C
C
C  ICHNL IS THE CHANNEL NUMBER; = 1 FOR THE HUB REGION,
C                               = NSLICE AT THE TIP
C
C
C-- LOCATE INPUT DATA FOR THIS CHANNEL AND STORE IT IN WORKING ARRAYS.
C
C I1 IS THE STARTING POINT IN THE INDCHN ARRAY FOR THIS CHANNEL
C
      I1 = INDCHN(ICHNL)
      IF ( ICHNL.NE.INDCHN(I1)) GO TO 290
C
C-- IF ABOVE TEST IS TRUE, THEN THE DATA IS NOT STORED WHERE EXPECTED
C
10    CONTINUE
      IFILM = INDCHN(I1+1)
      ICOR = INDCHN(I1+2)
      NFWD = INDCHN(I1+3)
      NSTA = INDCHN(I1+4)
      ISBLOK = INDCHN(I1+5)
      NBLKSZ = INDCHN(I1+6)
      IPLOT = INDCHN(I1+7)
      MD1 = INDCHN(I1+8)
      MD2 = INDCHN(I1+9)
      IHUB = INDCHN(I1+12)
      ITIP = INDCHN(I1+13)
      IN1 = I1 + 14
      CD = CHANL(ISBLOK)
      ALPHA = CHANL(ISBLOK+1)
      BETA = CHANL(ISBLOK+2)
      DELTA = CHANL(ISBLOK+3)
      EPS = CHANL(ISBLOK+4)
      ADUMP = CHANL(ISBLOK+6)
      SPAN = CHANL(ISBLOK+7)
      S(ICHNL) = SPAN
      BTA  = CHANL(ISBLOK+8)
      DLTYME = CHANL(ISBLOK+9)
      TEPS = CHANL(ISBLOK+10)
      NODSF = 5*NFWD
      NODST = 5*NSTA
      I1 = ISBLOK + 14
      I3 = ISBLOK + 14 + 2*NODST
C******
12    CONTINUE
      DO 205 I = 1,NODST
      IM = I1 + I
      TAU(I) = CHANL(IM)
      A(I) = TAU(I)*SPAN
```

```
      IM = IM + NODST                                                   NPREPAT 4081
205   DLX(I) = CHANL(IM)                                                NPREPAT 4082
      DO 215 I = 1,NSTA                                                 NPREPAT 4083
      IFLU = 5*I                                                        NPREPAT 4084
      DH(I) = 4.0*A(IFLU)/(2.*(SPAN+TAU(IFLU)))                         NPREPAT 4085
      IM = I3 + I                                                       NPREPAT 4086
      DHJ(I) = CHANL(IM)                                                NPREPAT 4087
      IM = IM + NSTA                                                    NPREPAT 4088
      DHF(I) = CHANL(IM)                                                NPREPAT 4089
      IM = IM + NSTA                                                    NPREPAT 4090
      XN(I) = CHANL(IM)                                                 NPREPAT 4091
14    CONTINUE                                                          NPREPAT 4092
      IF (DHJ(I).GT.0.0.AND.XN(I).GT.0.0) GO TO 202                     NPREPAT 4093
      AJET(I) = 0.0                                                     NPREPAT 4094
      GO TO 212                                                         NPREPAT 4095
202   CONTINUE                                                          NPREPAT 4096
      XOD=XN(I)/DHJ(I)                                                  NPREPAT 4097
      IF(XOD.LT.3.1.OR.XOD.GT.12.5)WRITE(6,527)I,XOD                    NPREPAT 4098
527   FORMAT(1H0,' WARNING, RATIO OF JET HOLE SPACING TO JET DIAMETER   NPREPAT 4099
     1'FOR JET ',I2,' IS',F10.4,' WHICH IS OUT OF'/                     NPREPAT 4100
     2'  THE RANGE OF VALIDITY OF THE CORRELATION.')                    NPREPAT 4101
      CNUM(I)=SPAN/XN(I)                                                NPREPAT 4102
C  CNUM IS THE NUMBER OF IMPINGEMENT JETS AT CHANNEL NODE I             NPREPAT 4103
C  TOTAL JET AREA IS (AREA OF ONE JET)*(NUMBER OF JETS)                 NPREPAT 4104
C                                                                       NPREPAT 4105
54    AJET(I)=3.14159*DHJ(I)**2/4.*CNUM(I)                              NPREPAT 4106
212   CONTINUE                                                          NPREPAT 4107
      IM = IM + NSTA                                                    NPREPAT 4108
      RR(I) = CHANL(IM)                                                 NPREPAT 4109
      IM = IM + NSTA                                                    NPREPAT 4110
      DP(I) = CHANL(IM)                                                 NPREPAT 4111
      IM = IM + NSTA                                                    NPREPAT 4112
      SP(I) = CHANL(IM)                                                 NPREPAT 4113
      IM = IM + NSTA                                                    NPREPAT 4114
      IM = IM + NSTA                                                    NPREPAT 4115
      INN = IN1 + I                                                     NPREPAT 4116
      IHC(I) = INDCHN(INN)                                              NPREPAT 4117
215   CONTINUE                                                          NPREPAT 4118
C******                                                                 NPREPAT 4119
C                                                                       NPREPAT 4120
C-- NOW, GIVEN SLICE, ICHNL, EVALUATE B.C. AT METAL NODE POINTS. IN THE NPREPAT 4121
C   FOLLOWING:                                                          NPREPAT 4122
C--   XS & XP ARE DISTANCE FROM LEADING EDGE, ALONG OUTSIDE SURFACE     NPREPAT 4123
C     (INCHES), FOR SUCTION & PRESSURE SIDES.                           NPREPAT 4124
C-- THE CONVENTIONS USED IN THE FOLLOWING ARE:  INDEX BEGINING WITH -I- NPREPAT 4125
C-- IS A SLICE INDEX, INDEX BEGINING WITH -N- IS A TIME INDEX, INDEX    NPREPAT 4126
C-- BEGINING WITH -L- IS A N X INDEX, AND AN INDEX BEGINING WITH -J- IS NPREPAT 4127
C-- A PROPERTY INDEX I.E. HG,QG,TG,PG.                                  NPREPAT 4128
C                                                                       NPREPAT 4129
C-- FIRST, CHECK THAT THIS IS A TRANSIENT CASE, AND DETERMINE THE MAX.  NPREPAT 4130
C-- BCTIME INDEX, NMX.                                                  NPREPAT 4131
      NMX = 1                                                           NPREPAT 4132
310   IF (BCTIME(NMX+1).LE.0.0) GO TO 315                               NPREPAT 4133
      NMX = NMX + 1                                                     NPREPAT 4134
      GO TO 310                                                         NPREPAT 4135
315   CONTINUE                                                          NPREPAT 4136
C                                                                       NPREPAT 4137
C-- NOW, IF THIS IS A TRANSIENT, FIND THE LOCATION IN THE BCTIME ARRAY  NPREPAT 4138
C-- OF THE CURRENT TIME, AND CALCULATE THE VALUE OF THE INTERPOLATING   NPREPAT 4139
C-- PARAMETER, TMFRAC.                                                  NPREPAT 4140
```

```
      TMFRAC = 0.0                                                      NPREPAT 4141
      NLST = 1                                                          NPREPAT 4142
      IF (NMX.EQ.1.OR.TYME.LE.0.0) GO TO 330                            NPREPAT 4143
C-- THE ABOVE TRANSFER OCCURS IF THIS IS A STEADY STATE PROBLEM         NPREPAT 4144
C-- THE FOLLOWING TRANSFER OCCURS IF WE ARE BEYOND THE LAST BCTIME ENTRYNPREPAT 4145
      NLST = NMX                                                        NPREPAT 4146
      IF (TYME.GE.BCTIME(NMX)) GO TO 330                                NPREPAT 4147
      NMXM1 = NMX - 1                                                   NPREPAT 4148
      DO 320 N = 1,NMXM1                                                NPREPAT 4149
      NLST = N                                                          NPREPAT 4150
      IF (TYME.GE.BCTIME(N).AND.TYME.LT.BCTIME(N+1)) GO TO 325          NPREPAT 4151
320   CONTINUE                                                          NPREPAT 4152
325   TMFRAC = (TYME-BCTIME(NLST))/(BCTIME(NLST+1)-BCTIME(NLST))        NPREPAT 4153
C                                                                       NPREPAT 4154
C-- NEXT, SEARCH THE BCXS & BCXP ARRAYS TO FIND THE X INTERPOLATING     NPREPAT 4155
C-- FACTORS, XSF & XPF, FOR POSITIONS XS & XP, SLICE ICHNL.             NPREPAT 4156
C-- THE BRACKETING INDICES ARE LBLOWS & LABOVS AND LBLOWP & LABOVP.     NPREPAT 4157
C                                                                       NPREPAT 4158
C-- THE STARTING POINTS IN THE BCXS & BCXP ARRAYS FOR THIS SLICE ARE:   NPREPAT 4159
C                                                                       NPREPAT 4160
330   LSS = (ICHNL-1)*NBCS                                              NPREPAT 4161
      LSP = (ICHNL-1)*NBCP                                              NPREPAT 4162
      XS = 0.0                                                          NPREPAT 4163
      XP = 0.0                                                          NPREPAT 4164
C                                                                       NPREPAT 4165
C-- THE STARTING POINTS IN THE PROPERTY ARRAYS, FOR THE LATEST TIME STEPNPREPAT 4166
C-- ARE GIVEN BY:                                                       NPREPAT 4167
C                                                                       NPREPAT 4168
      JS1S = NSLICE*(NLST-1)*NBCS + LSS                                 NPREPAT 4169
      JS1P = NSLICE*(NLST-1)*NBCP + LSP                                 NPREPAT 4170
      IF (NMX.EQ.NLST) GO TO 335                                        NPREPAT 4171
      JS2S = JS1S + NSLICE*NBCS                                         NPREPAT 4172
      JS2P = JS1P + NSLICE*NBCP                                         NPREPAT 4173
335   HG(1) = BCHGS(JS1S+1) + TMFRAC*(BCHGS(JS2S+1)-BCHGS(JS1S+1))      NPREPAT 4174
      TG(1) = BCTGS(JS1S+1) + TMFRAC*(BCTGS(JS2S+1)-BCTGS(JS1S+1)) +460.NPREPAT 4175
      QG(1) = BCQGS(JS1S+1) + TMFRAC*(BCQGS(JS2S+1)-BCQGS(JS1S+1))      NPREPAT 4176
      PG(1) = BCPGS(JS1S+1) + TMFRAC*(BCPGS(JS2S+1)-BCPGS(JS1S+1))      NPREPAT 4177
C                                                                       NPREPAT 4178
      DO 350 K = 2,NSTA,2                                               NPREPAT 4179
C-- THE OUTSIDE SURFACE NODE NUMBERS FOR S & P SIDES ARE:               NPREPAT 4180
      NNS = 5*K - 4                                                     NPREPAT 4181
      NNP = 5*K + 1                                                     NPREPAT 4182
      XS = XS + DLX(NNS)                                                NPREPAT 4183
      XP = XP + DLX(NNP)                                                NPREPAT 4184
C                                                                       NPREPAT 4185
      DO 340 L = 1,NBCS                                                 NPREPAT 4186
      LBLOWS = LSS + L -1                                               NPREPAT 4187
      LABOVS = LSS + L                                                  NPREPAT 4188
      IF (BCXS(LABOVS).GT.XS) GO TO 342                                 NPREPAT 4189
340   CONTINUE                                                          NPREPAT 4190
C                                                                       NPREPAT 4191
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXS TABLE     NPREPAT 4192
C                                                                       NPREPAT 4193
342   XSF = (XS-BCXS(LBLOWS))/(BCXS(LABOVS)-BCXS(LBLOWS))               NPREPAT 4194
C                                                                       NPREPAT 4195
      DO 345 L = 1,NBCP                                                 NPREPAT 4196
      LBLOWP = LSP + L - 1                                              NPREPAT 4197
      LABOVP = LSP + L                                                  NPREPAT 4198
      IF (BCXP(LABOVP).GT.XP) GO TO 347                                 NPREPAT 4199
345   CONTINUE                                                          NPREPAT 4200
```

```
C
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXP TABLE
C
347   XPF = (XP-BCXP(LBLOWP))/(BCXP(LABOVP)-BCXP(LBLOWP))
C
C-- NOW THE FRACTIONS ARE KNOWN, CALCULATE THE INTERPOLATED PROPERTIES.
C-- FIRST, FOR THE STEADY STATE OR FOR TIMES BEYOND THE LAST BCTIME:
C
      JB1S = NSLICE*(NLST-1)*NBCS + LBLOWS
      JB1P = NSLICE*(NLST-1)*NBCP + LBLOWP
      HG(K) = BCHGS(JB1S) + XSF*(BCHGS(JB1S+1)-BCHGS(JB1S))
      HG(K+1) = BCHGP(JB1P) + XPF*(BCHGP(JB1P+1)-BCHGP(JB1P))
      QG(K) = BCQGS(JB1S) + XSF*(BCQGS(JB1S+1)-BCQGS(JB1S))
      QG(K+1) = BCQGP(JB1P) + XPF*(BCQGP(JB1P+1)-BCQGP(JB1P))
      TG(K) = BCTGS(JB1S) + XSF*(BCTGS(JB1S+1)-BCTGS(JB1S)) + 460.
      TG(K+1) = BCTGP(JB1P) + XPF*(BCTGP(JB1P+1)-BCTGP(JB1P)) + 460.
      PG(K) = BCPGS(JB1S) + XSF*(BCPGS(JB1S+1)-BCPGS(JB1S))
      PG(K+1) = BCPGP(JB1P) + XPF*(BCPGP(JB1P+1)-BCPGP(JB1P))
C
      IF (NMX.EQ.1.OR.TYME.GE.BCTIME(NMX).OR.TYME.LE.0.0) GO TO 350
      JB2S = NSLICE*NLST*NBCS + LBLOWS
      JB2P = NSLICE*NLST*NBCP + LBLOWP
      AHG = BCHGS(JB2S) + XSF*(BCHGS(JB2S+1)-BCHGS(JB2S))
      HG(K) = HG(K) + TMFRAC*(AHG-HG(K))
      AHG = BCHGP(JB2P) + XPF*(BCHGP(JB2P+1)-BCHGP(JB2P))
      HG(K+1) = HG(K+1) + TMFRAC*(AHG-HG(K+1))
      AQG = BCQGS(JB2S) + XSF*(BCQGS(JB2S+1)-BCQGS(JB2S))
      QG(K) = QG(K) + TMFRAC*(AQG-QG(K))
      AQG = BCQGP(JB2P) + XPF*(BCQGP(JB2P+1)-BCQGP(JB2P))
      QG(K+1) = QG(K+1) + TMFRAC*(AQG-QG(K+1))
      ATG = BCTGS(JB2S) + XSF*(BCTGS(JB2S+1)-BCTGS(JB2S)) + 460.
      TG(K) = TG(K) + TMFRAC*(ATG-TG(K))
      ATG = BCTGP(JB2P) + XPF*(BCTGP(JB2P+1)-BCTGP(JB2P)) + 460.
      TG(K+1) = TG(K+1) + TMFRAC*(ATG-TG(K+1))
      APG = BCPGS(JB2S) + XSF*(BCPGS(JB2S+1)-BCPGS(JB2S))
      PG(K) = PG(K) + TMFRAC*(APG-PG(K))
      APG = BCPGP(JB2P) + XPF*(BCPGP(JB2P+1)-BCPGP(JB2P))
      PG(K+1) = PG(K+1) + TMFRAC*(APG-PG(K+1))
350   CONTINUE
      RETURN
290   WRITE(6,295) ICHNL
295   FORMAT(//' CHANNEL NO. ',I3,' DATA STORAGE IS MESSED UP',
     Z         ' *%><;:%%*%&@$?***%****')
149   CONTINUE
      RETURN
      END


C----SOURCE.NTARAYT
      SUBROUTINE TARRAY(JS,JSENS,DELTAN)
C
C-    SOURCE.NTARAYT----
C
C
C+++++ A SUBROUTINE TO SET UP THE COEFFICIENT ARRAY TO SOLVE FOR
C        BLADE TEMPERATURES,  TRANSIENT CALCULATIONS.
C      FIRST PUT TOGETHER FROM STEADY STATE PROGRAM, NOVEMBER 24,1975
C
C
C************************************************************************
C************************************************************************
C
```

```
C                                                                       NTARAYT 4261
C                                                                       NTARAYT 4262
C  NODE NAMES FOR STATION IS ARE:                                       NTARAYT 4263
C    LCOOL = COOLANT NODE AT IS.                                        NTARAYT 4264
C    LIN   = INSIDE WALL NODE.                                          NTARAYT 4265
C    L     = MID WALL NODE.                                             NTARAYT 4266
C    LOUT  = OUTSIDE WALL NODE.                                         NTARAYT 4267
C    LCUP  = ADJACENT UPSTREAM COOLANT NODE.                            NTARAYT 4268
C    LCUPS = ADJACENT INSIDE WALL NODE                                  NTARAYT 4269
C    LUP   = ADJACENT UPSTREAM MID-WALL NODE.                           NTARAYT 4270
C    LDN   = ADJACENT DOWNSTREAM MID-WALL NODE.                         NTARAYT 4271
C    LJ    = JUNCTION OF COATING AND WALL METAL                         NTARAYT 4272
C                                                                       NTARAYT 4273
C                                                                       NTARAYT 4274
C                                 IS                                    NTARAYT 4275
C                                  |                                    NTARAYT 4276
C                ------------LOUT-*-------------                        NTARAYT 4277
C                --------------LJ *-------------                        NTARAYT 4278
C                ///////////////////////////////                        NTARAYT 4279
C           WALL ///LUP-*///////L-*/////LDN-*///                        NTARAYT 4280
C                ///////////////////////////////                        NTARAYT 4281
C                -LCUPS-*-----LIN-*-------------                        NTARAYT 4282
C           FLOW                                                        NTARAYT 4283
C           --->   LCUP-*    LCOOL-*                                    NTARAYT 4284
C                                                                       NTARAYT 4285
C                -LCUPP-*--LCOOLP-*-------------                        NTARAYT 4286
C                                                                       NTARAYT 4287
C                 PRESSURE SIDE WALL OR PLENUM                          NTARAYT 4288
C                                                                       NTARAYT 4289
C***********************************************************************NTARAYT 4290
C***********************************************************************NTARAYT 4291
C                                                                       NTARAYT 4292
      REAL*8 TCOF                                                       NTARAYT 4293
      COMMON /BOUND/ BCXS(400),   BCXP(400),   BCHGS(1000), BCHGP(1000),NTARAYT 4294
     Z               BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),NTARAYT 4295
     Z               BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),   NTARAYT 4296
     Z               QHUBIN(400), QHUB(80),    TTIPIN(400), TTIP(80),   NTARAYT 4297
     Z               QTIPIN(400), QTIP(80),    RHOVG(400),  PEX(400),   NTARAYT 4298
     Z               BCTIME(50),  TTIO(50),    PTIO(50),    WPLEN,      NTARAYT 4299
     Z               WSVST(50),   AKCTBL(20),  AKWTBL(20),  NBCS,   NBCP NTARAYT 4300
C                                                                       NTARAYT 4301
      COMMON /FLMCOL/ RHOVGA(80), PG(80),    XFC(80),   FLMEFF(80),     NTARAYT 4302
     Z                XMUC(80),   EMES(80),  REFC(80),  NFCSUP(80)      NTARAYT 4303
C                                                                       NTARAYT 4304
      COMMON /MATRX/ TCOF(400,30)                                       NTARAYT 4305
C                                                                       NTARAYT 4306
      COMMON /PRPS/ CPO,      GAMO,     DP(80),     SP(80),   RE(80),   NTARAYT 4307
     Z              CPC(80),  GAMC(80), DUMR1(80),  DUMR2(80)           NTARAYT 4308
C                                                                       NTARAYT 4309
      COMMON /TCO/ ADUMP,     BTA,       CD,          CP,               NTARAYT 4310
     Z             GAM,       PIM,       R,           SPAN,        TOG, NTARAYT 4311
     Z             WDUMP,     WIM,       AKC(15,80),  AKW(15,80),       NTARAYT 4312
     Z             A(400),    AJET(80),  AM2(80),     CNUM(80),         NTARAYT 4313
     Z             DH(80),    DHF(80),   DHJ(80),                       NTARAYT 4314
     Z             DLX(400),  FF(80),    HC(80),      HG(80),           NTARAYT 4315
     Z             P(2,15,80),PEXIT(15), PUMP(80),    QG(80),           NTARAYT 4316
     Z             QSNK(80),  RR(80),    S(15),       T(2,15,400),      NTARAYT 4317
     Z             TG(80),    TAU(400),  WFC(80),                       NTARAYT 4318
     Z             WJ(15,80), WCROS(2,15,80),         XN(80),           NTARAYT 4319
     Z             ICOR,      IFILM,     IHUB,        ITIP,             NTARAYT 4320
```

```
     Z               ISBLOK,     ISLICE,     NBLKSZ,     NSLICE,
     Z               NFWD,       NSTA,       IHC(80)
C
      COMMON /TRNSNT/ RHOC,       RHOM,       SPHTC,      SPHTM,
     Z                DLTYME,     TYME,       TEPS,       TYMMAX
      DIMENSION EFAREA(80), DELTAN(15)
C
      RTRNVM = 0.
      RCHRDM = 0.
      TREPS = 1.0
      IF (TYME.GE.0.) TREPS = TEPS
      HX = 1.0
      RCVRY = .89
300   CONTINUE
      SPAN = S(ISLICE)
      NODST = 5*NSTA
      NODSF = 5*NFWD
      DO 308 J = 1,30
      DO 308 I = 1,400
308   TCOF(I,J) = 0.0
      DO 309 I = 1,9
      ICO = NODST + I
309   DLX(ICO) = 0.0
      ICOMS = JS + 2 - 4*JSENS
      ICOMP = JS - 2 + 4*JSENS
C****ICOMS IS STATION ADJACENT TO JS, IN SUCTION DIRECTION
C****ICOMP IS STATION ADJACENT TO JS,IN PRESSURE DIRECTION
      IF (ICOMS.LT.0) ICOMS=2
      IF (ICOMP.LT.1) ICOMP=1
310   CONTINUE
C
C**   BEGIN OVERALL LOOP, WHERE LOOP VARIABLE (IS) IS THE STATION NUMBER
C
      DO 440 IS = 1,NSTA
      ISUP   = IS - 2
      YIMP   = 0.0
      YFINS  = 0.0
      YCONV  = 0.0
      YIMPU  = 0.0
      YFINSU = 0.0
      YCONVU = 0.0
      YIMPUU = 0.0
      YFNSUU = 0.0
      YCNVUU = 0.0
      IF (IHC(IS).EQ.1) YIMP=1.0
      IF (IHC(IS).EQ.2) YCONV= (1.0+RCVRY*AM2(IS)*(GAMC(IS)-1.)/2.)
      IF (IHC(IS).EQ.3) YFINS=1.0
      FACTOR = 1.0
      IF (IS.EQ.ICOMS.OR.IS.EQ.ICOMP) FACTOR = .5
      ISENS = 0
      ISEN = IS - 2*(IS/2)
C
C  IF (IS.GT.NFWD+1)-- IN TRAILING EDGE REGION, GO TO 380
C
      IF(IS.GT.NFWD+1) GO TO 380
C
      J9 = 16
      LCOOL = 5*IS
      LIN   =LCOOL-1
      L     =LCOOL-2
```

```
      LJ    =LCOOL - 3                                                  NTARAYT
      LOUT  =LCOOL-4                                                    NTARAYT
      IF (IHC(IS).NE.2) GO TO 320                                       NTARAYT
C                                                                       NTARAYT
C    FORCED CONVECTION HC:                                              NTARAYT
      TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.                    NTARAYT
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)                                NTARAYT
      RE(IS) = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU)  NTARAYT
      HC(IS) = .023*12.*(C/DH(IS))*(RE(IS)**.8)*(PD**.333)              NTARAYT
C                                                                       NTARAYT
320   CONTINUE                                                          NTARAYT
      IF (IS.GE.JS) GO TO 322                                           NTARAYT
C                                                                       NTARAYT
C   SPECIAL CASE FOR STATION NUMBER 1:                                  NTARAYT
      IF(IS.EQ.1) ISUP = 2                                              NTARAYT
C                                                                       NTARAYT
C  IF STATION IS IS FORWARD OF FLOW SPLIT, AND ON SAME SIDE, GO TO 370  NTARAYT
      IF (ISEN.EQ.JSENS) GO TO 370                                      NTARAYT
C                                                                       NTARAYT
322   CONTINUE                                                          NTARAYT
      IF (ISUP.LT.1) ISUP = 1                                           NTARAYT
      IF (IHC(ISUP).EQ.1) YIMPU = 1.0                                   NTARAYT
      IF (IHC(ISUP).EQ.2) YCONVU=1.0+RCVRY*AM2(ISUP)*(GAMC(ISUP)-1.)/2. NTARAYT
      IF (IHC(ISUP).EQ.3) YFINSU = 1.0                                  NTARAYT
      LCUP = LCOOL - 10                                                 NTARAYT
      LCUPS = LCOOL - 11                                                NTARAYT
      LUP = LCOOL - 12                                                  NTARAYT
      LDN = LCOOL + 8                                                   NTARAYT
      IF (IS.NE.2) GO TO 324                                            NTARAYT
C                                                                       NTARAYT
C  IF THIS IS STATION NUMBER 2:                                         NTARAYT
      LCUP = 5                                                          NTARAYT
      LCUPS = 4                                                         NTARAYT
      LUP = 3                                                           NTARAYT
324   CONTINUE                                                          NTARAYT
      IF (IS.GT.1) GO TO 326                                            NTARAYT
C                                                                       NTARAYT
C  IF THIS IS STATION NUMBER 1:                                         NTARAYT
      LCUP = 10                                                         NTARAYT
      LCUPS = 9                                                         NTARAYT
      LUP = 8                                                           NTARAYT
326   CONTINUE                                                          NTARAYT
C                                                                       NTARAYT
C  IS = 1, STATION NO. 1,  LEADING EDGE NODES                           NTARAYT
C                                                                       NTARAYT
      IF (IS.NE.JS) GO TO 330                                           NTARAYT
C                                                                       NTARAYT
C***********************************************************************NTARAYT
C*********                                                              NTARAYT
C********* THIS BLOCK COMPUTES TCOF ELEMENTS FOR THE STATION AT WHICH   NTARAYT
C           THE FLOW SPLITS                                             NTARAYT
C*********  IS = JS                                                     NTARAYT
C*********                                                              NTARAYT
C***********************************************************************NTARAYT
C                                                                       NTARAYT
328   CONTINUE                                                          NTARAYT
      DX1  = DLX(LUP)                                                   NTARAYT
      IF (DX1.EQ.0.0) DX1  = DLX(L)                                     NTARAYT
      DX2  = DLX(LDN)                                                   NTARAYT
      IF (DX2.EQ.0.0) DX2  = DLX(L)                                     NTARAYT
```

```
      DX3   = DLX(LUP-2)
      IF (DX3.EQ.0.0) DX3   = DLX(LOUT)
      DX4   = DLX(LDN-2)
      IF (DX4.EQ.0.0) DX4   = DLX(LOUT)
      DX5   = DLX(LUP+1)
      IF (DX5.EQ.0.0) DX5   = DLX(LIN)
      DX6   = DLX(LDN+1)
      IF (DX6.EQ.0.0) DX6   = DLX(LIN)
      DX9   = DLX(LUP-1)
      IF (DX9.EQ.0.0) DX9   = DLX(LJ)
      DX10 = DLX(LDN-1)
      IF (DX10.EQ.0.) DX10 = DLX(LJ)
      CURV = 1.0 + (DX9+DX10)/(DX3+DX4)
C
C   CURV IS A MEASURE OF THE CURVATURE OF THE BLADE AT STATION IS.
C     CURV = 2.0 IS A STRAIGHT SECTION OF WALL,
C     CURV < 2.0 IS A CONVEX SECTION, AND
C     CURV > 2.0 IS A CONCAVE SECTION
C
      TRTRMC = 0.0
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMC=(3600./144.)*RHOC*SPHTC*
     Z    (TAU(LOUT)**2)/(4.*AKC(ISLICE,IS)*DLTYME)
      TCOF(LOUT,13) = -TREPS + TRTRMC*CURV
      TCOF(LOUT,12)=TRTRMC*CURV + TREPS*(1.0+(1.0-BTA)*HG(IS)*TAU(LOUT)/
     Z                                             (12.*AKC(ISLICE,IS)))
      TCOF(LOUT,J9) = -(1.-BTA)*TREPS*FLMEFF(IS)*HG(IS)*TAU(LOUT)/
     Z                                              (12.*AKC(ISLICE,IS))
      TCOF(LOUT,24) = (BTA*QG(IS) + (1.0-BTA)*HG(IS)*TG(IS)*
     Z           (1.0-FLMEFF(IS)))*TAU(LOUT)/(12.*AKC(ISLICE,IS))
     Z - T(1,ISLICE,LOUT)*((1.-TREPS)*((1.-BTA)*HG(IS)*TAU(LOUT)/
     Z           (12.*AKC(ISLICE,IS)) + 1.) - TRTRMC*CURV)
     Z + T(1,ISLICE,LJ)*(1.-TREPS+TRTRMC*CURV)
     Z + T(1,ISLICE,LCOOL)*(1.-TREPS)*FLMEFF(IS)*(1.-BTA)*HG(IS)*
     Z          TAU(LOUT)/(12.*AKC(ISLICE,IS))
C
      TCOF(LJ,11) = TREPS
      TCOF(LJ,13) = TREPS*(AKW(ISLICE,IS)/AKC(ISLICE,IS))*
     Z        (2.*TAU(LOUT)/TAU(L))*(DX1+DX2+DX3+DX4)/(DX9+DX10+DX3+DX4)
      TCOF(LJ,12) = - TCOF(LJ,11) - TCOF(LJ,13)
      TCOF(LJ,24) = (1.-TREPS)*((T(1,ISLICE,LJ)-T(1,ISLICE,LOUT)) +
     Z                    (T(1,ISLICE,LJ)-T(1,ISLICE,L))*TCOF(LJ,13)/TREPS)
C
      J1 = 12 - L + LUP
      J2 = 12 - L + LDN
C
      THETA1 = (DX1+DX2+DX5+DX6)/(DX1+DX2+DX9+DX10)
      THETA2 = ((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10)
      THETA3 = ((TAU(L)+TAU(LDN))/(2.*DX2))*2.*TAU(L)/(DX1+DX2+DX9+DX10)
      THETA6 = 24.*TAU(L)/(AKW(ISLICE,IS)*S(ISLICE)*(DX1+DX2+DX9+DX10))
      THETA4 = 0.0
      THETA5 = 0.0
      HUB1 = 0.0
      HUB3 = 0.0
      TIP1 = 0.0
      TIP3 = 0.0
      IF (ISLICE.EQ.1) GO TO 3290
C
C   FOR A SLICE THAT IS NOT AT THE HUB OF THE BLADE:
C
      THETA4 = (TAU(L)/S(ISLICE))*(TAU(L)/(S(ISLICE)+S(ISLICE-1)))*
```

```
     Z                      2.*(DX1+DX2)/(DX1+DX2+DX9+DX10)
      IF (ISLICE.EQ.NSLICE) GO TO 3292
      THETA5 = THETA4*(S(ISLICE)+S(ISLICE-1))/(S(ISLICE)+S(ISLICE+1))
      TBELOW = T(1,ISLICE-1,L)
      TABOVE = T(1,ISLICE+1,L)
      GO TO 3294
C
C  FOR THE SLICE AT THE HUB END OF THE BLADE:
C
3290  CONTINUE
      IF (IHUB.EQ.1) HUB1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*
     Z                                             (2.*(TAU(L)/S(1))**2)
C  FOR IHUB = 1, HUB TEMPERATURE IS SPECIFIED**************************
      THETA5 = 0.0
      IF (NSLICE.GT.1) THETA5 = (TAU(L)/S(1))*(TAU(L)/(S(1)+S(2)))*
     Z                               (2.*(DX1+DX2)/(DX1+DX2+DX9+DX10))
      IF (IHUB.EQ.3) HUB3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/
     Z                                   (AKW(1,IS)*12.*S(1))
C IHUB = 3, THE HEAT FLUX AT THE HUB END IS SPECIFIED (BTU/HR FT**2) ***
      TBELOW = T(1,1,L)
      IF (IHUB.EQ.1)   TBELOW = THUB(IS)
      TABOVE = T(1,1,L)
      IF (NSLICE.GT.1) TABOVE = T(1,2,L)
      IF (NSLICE.GT.1) GO TO 3294
C
C  FOR THE SLICE AT THE BLADE TIP, (IF THERE ARE MORE THAN 1 SLICES
C      BEING CONSIDERED) : ********************************
C
3292  CONTINUE
      IF (ITIP.EQ.1) TIP1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*
     Z                               (2.*(TAU(L)/S(NSLICE))**2)
      IF (NSLICE.GT.1) TBELOW = T(1,ISLICE-1,L)
      IF (ITIP.EQ.3) TIP3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/
     Z                               (AKW(NSLICE,IS)*12.*S(NSLICE))
      TABOVE = T(1,ISLICE,L)
      IF (ITIP.EQ.1) TABOVE = TTIP(IS)
C
C
3294  CONTINUE
      THETA9 = 0.0
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) THETA9 = 2.*3600.*RHOM*SPHTM*
     Z                 (DX1+DX2)*(TAU(L)**2)/(144.*AKW(ISLICE,IS)*
     Z                                 (DX1+DX2+DX9+DX10)*DLTYME)
C
      TCOF(L,11)  = 1.0*TREPS
      TCOF(L,13)  = THETA1*TREPS
      TCOF(L,J1)  = THETA2*TREPS
      TCOF(L,J2)  = THETA3*TREPS
      TCOF(L,12)  = (-1.0 - THETA1 - THETA2 - THETA3 - THETA4 - THETA5 -
     Z                          HUB1 - TIP1)*TREPS - THETA9
      TCOF(L,24)  = QSNK(IS)*THETA6 - (THETA4+HUB1)*TBELOW
     Z           -(THETA5+TIP1)*TABOVE - QHUB(IS)*HUB3 + QTIP(IS)*TIP3
     Z           -(1-TREPS)*(T(1,ISLICE,LUP)*THETA2+T(1,ISLICE,LJ)
     Z                 +T(1,ISLICE,LIN)*THETA1+T(1,ISLICE,LDN)*THETA3)
     Z           +T(1,ISLICE,L)*((1.0-TREPS)*(1.+THETA1+THETA2+THETA3
     Z                             +THETA4+THETA5+HUB1+TIP1) - THETA9)
C
C
      AHTRN1 = (DX5 + DX6)*S(ISLICE)/2.
      THETA8 = 2.*HX*HC(IS)*AHTRN1*TAU(L)/(12.*AKW(ISLICE,IS)*S(ISLICE)
```

```
     Z                                  *(DX1+DX2+DX5+DX6))             NTARAYT 4561
      TCOF(LIN,11) =  TREPS                                             NTARAYT 4562
      TCOF(LIN,24) = - THETA8*TOG - (1.0-TREPS)*(T(1,ISLICE,L)          NTARAYT 4563
     Z                                      -T(1,ISLICE,LIN)*(1.0+THETA8))  NTARAYT 4564
      TCOF(LIN,12) = (-1.0 - THETA8)*TREPS                              NTARAYT 4565
C                                                                       NTARAYT 4566
      TCOF(LCOOL,12) = 1.0                                              NTARAYT 4567
      TCOF(LCOOL,24) = TOG                                              NTARAYT 4568
      IF (TYME.GT.0.) RCHRD = (144./3600.)*AKW(ISLICE,IS)*DLTYME/       NTARAYT 4569
     Z                         (RHOM*SPHTM*(((DX1+DX2)/2.)**2))         NTARAYT 4570
      IF (TYME.GT.0.) RTRNV = (144./3600.)*AKW(ISLICE,IS)*DLTYME/       NTARAYT 4571
     Z                         (RHOM*SPHTM*(TAU(L)**2))                 NTARAYT 4572
      IF (RCHRD.GT.RCHRDM) RCHRDM = RCHRD                               NTARAYT 4573
      IF (RTRNV.GT.RTRNVM) RTRNVM = RTRNV                               NTARAYT 4574
C                                                                       NTARAYT 4575
      GO TO 440                                                         NTARAYT 4576
C                                                                       NTARAYT 4577
C                                                                       NTARAYT 4578
C                                                                       NTARAYT 4579
C                                                                       NTARAYT 4580
C                                                                       NTARAYT 4581
C                                                                       NTARAYT 4582
C                                                                       NTARAYT 4583
C                                                                       NTARAYT 4584
C                                                                       NTARAYT 4585
C***********************************************************************NTARAYT 4586
C********                                                               NTARAYT 4587
C******** THIS BLOCK COMPUTES TCOF ELEMENTS FOR GENERAL FLOW STATIONS    NTARAYT 4588
C********                                                               NTARAYT 4589
C***********************************************************************NTARAYT 4590
C                                                                       NTARAYT 4591
330   CONTINUE                                                          NTARAYT 4592
      DX1 = DLX(L)                                                      NTARAYT 4593
      DX2 = DLX(LDN)                                                    NTARAYT 4594
      DX3 = DLX(LOUT)                                                   NTARAYT 4595
      DX4 = DLX(LDN-2)                                                  NTARAYT 4596
      DX5 = DLX(LIN)                                                    NTARAYT 4597
      DX6 = DLX(LDN+1)                                                  NTARAYT 4598
      DX7 = DLX(LCOOL)                                                  NTARAYT 4599
      DX9 = DLX(LJ)                                                     NTARAYT 4600
      DX10 = DLX(LDN-1)                                                 NTARAYT 4601
      IF (IS.GT.1) GO TO 332                                            NTARAYT 4602
C                                IF THIS IS STATION NUMBER 1:           NTARAYT 4603
      DX1 = DLX(LUP)                                                    NTARAYT 4604
      DX9 = DLX(LUP-1)                                                  NTARAYT 4605
      DX3 = DLX(LUP-2)                                                  NTARAYT 4606
      DX5 = DLX(LUP+1)                                                  NTARAYT 4607
      DX7 = DLX(LUP+2)                                                  NTARAYT 4608
C                                                                       NTARAYT 4609
332   CONTINUE                                                          NTARAYT 4610
      LCUPP  = LCOOL - 5                                                NTARAYT 4611
      LCOOLP = LCOOL + 4                                                NTARAYT 4612
      J1 = 12 - L + LUP                                                 NTARAYT 4613
      J2 = 12 - L + LDN                                                 NTARAYT 4614
      J4 = 12 - LCOOL + LCUPS                                           NTARAYT 4615
      J5 = 12 - LCOOL + LCUP                                            NTARAYT 4616
      J6 = 12 - LCOOL + LCUPP                                           NTARAYT 4617
      J8 = 12 - LCOOL + LCOOLP                                          NTARAYT 4618
C                                                                       NTARAYT 4619
333   CONTINUE                                                          NTARAYT 4620
```

```
      AHTRN1 = (DX5 + DX6)*S(ISLICE)/2.                                 NTARAYT 4621
      A1 = SPAN*DX5/2.                                                  NTARAYT 4622
      A2 = A1                                                           NTARAYT 4623
      A3 = 0.0                                                          NTARAYT 4624
      A4 = 0.0                                                          NTARAYT 4625
C                                                                       NTARAYT 4626
      IF (IHC(IS).EQ.3.AND.IS.LE.NFWD)                                  NTARAYT 4627
     Z        CALL HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EFAREA)  NTARAYT 4628
      IF (IHC(IS-2).EQ.3) A1      = EFAREA(IS-2)/2.                     NTARAYT 4629
      IF (IHC(IS).EQ.3)   A2      = EFAREA(IS)/2.                       NTARAYT 4630
      IF (IHC(IS).EQ.3)   AHTRN1 = EFAREA(IS)                           NTARAYT 4631
C                                                                       NTARAYT 4632
      I3 = 12 - LIN + LCOOL                                             NTARAYT 4633
340   CONTINUE                                                          NTARAYT 4634
C                                                                       NTARAYT 4635
      CURV = 1.0 + (DX9+DX10)/(DX3+DX4)                                 NTARAYT 4636
C                                                                       NTARAYT 4637
C***********************************************************************NTARAYT 4638
C                                                                       NTARAYT 4639
C  FOR MID-METAL NODE (L) :                         IN GENERAL; I3 = 13NTARAYT 4640
C      TCOF(L,12), 12 REFERS TO NODE L                             J1 =  2NTARAYT 4641
C           (L,J1), J1 REFERS TO NODE LUP                          J2 = 22NTARAYT 4642
C           (L,J2), J2 REFERS TO NODE LDN                          J4 =  1NTARAYT 4643
C           (L,11), 11 REFERS TO NODE LJ                           J5 =  2NTARAYT 4644
C           (L,13), 13 REFERS TO NODE LIN                          J6 =  6NTARAYT 4645
C  FOR THE COOLANT NODE (LCOOL) :                                  J8 = 16NTARAYT 4646
C  TCOF(LCOOL,12), 12 REFERS TO NODE LCOOL                              NTARAYT 4647
C      (LCOOL,J4), J4 REFERS TO NODE LCUPS = LUP+1                      NTARAYT 4648
C            ,J5), J5 REFERS TO NODE LCUP                               NTARAYT 4649
C            ,J6), J6 REFERS TO NODE LCUPP, (TRAILING EDGE REGION ONLY) NTARAYT 4650
C            ,11), 11 REFERS TO NODE LIN                                NTARAYT 4651
C            ,J8), J8 REFERS TO NODE LCOOLP, (TRAILING EDGE REGION ONLY)NTARAYT 4652
C                                                                       NTARAYT 4653
C***********************************************************************NTARAYT 4654
C                                                                       NTARAYT 4655
C FOR OUTSIDE NODE:                                                     NTARAYT 4656
C                                                                       NTARAYT 4657
      TRTRMC = 0.0                                                      NTARAYT 4658
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMC =                        NTARAYT 4659
     Z (3600./144.)*RHOC*SPHTC*(TAU(LOUT)**2)/(4.*AKC(ISLICE,IS)*DLTYME)NTARAYT 4660
      TCOF(LOUT,13) = -TREPS + TRTRMC*CURV                              NTARAYT 4661
      TCOF(LOUT,12) = (1.0 + (1.0-BTA)*HG(IS)*TAU(LOUT)/                NTARAYT 4662
     Z                      (12.*AKC(ISLICE,IS)))*TREPS + TRTRMC*CURV   NTARAYT 4663
      TCOF(LOUT,J9) = -(1.-BTA)*TREPS*FLMEFF(IS)*HG(IS)*TAU(LOUT)/      NTARAYT 4664
     Z                                          (12.*AKC(ISLICE,IS))    NTARAYT 4665
      TCOF(LOUT,24) = (BTA*QG(IS) + (1.0-BTA)*HG(IS)*TG(IS)*            NTARAYT 4666
     Z            (1.0-FLMEFF(IS)))*TAU(LOUT)/(12.*AKC(ISLICE,IS))      NTARAYT 4667
     Z - T(1,ISLICE,LOUT)*((1.-TREPS)*((1.-BTA)*HG(IS)*TAU(LOUT)/       NTARAYT 4668
     Z   (12.*AKC(ISLICE,IS)) + 1.) - TRTRMC*CURV)                      NTARAYT 4669
     Z + T(1,ISLICE,LJ)*(1.-TREPS+TRTRMC*CURV)                          NTARAYT 4670
     Z + T(1,ISLICE,LCOOL)*(1.-TREPS)*FLMEFF(IS)*(1.-BTA)*HG(IS)*       NTARAYT 4671
     Z    TAU(LOUT)/(12.*AKC(ISLICE,IS))                                NTARAYT 4672
C                                                                       NTARAYT 4673
C                                                                       NTARAYT 4674
C  AT JUNCTION OF COATING AND METAL, NODE LJ:                           NTARAYT 4675
C                                                                       NTARAYT 4676
      TCOF(LJ,11) = TREPS                                               NTARAYT 4677
      TCOF(LJ,13) = TREPS*(AKW(ISLICE,IS)/AKC(ISLICE,IS))*              NTARAYT 4678
     Z    (2.*TAU(LOUT)/TAU(L))*(DX1+DX2+DX3+DX4)/(DX9+DX10+DX3+DX4)    NTARAYT 4679
      TCOF(LJ,12) = - TCOF(LJ,11) - TCOF(LJ,13)                         NTARAYT 4680
```

```
      TCOF(LJ,24) = (1.-TREPS)*((T(1,ISLICE,LJ)-T(1,ISLICE,LOUT)) +
     Z        (T(1,ISLICE,LJ)-T(1,ISLICE,L))*TCOF(LJ,13)/TREPS)
C
C
C FOR MID-METAL NODE:
C
      THETA1 = (DX1+DX2+DX5+DX6)/(DX1+DX2+DX9+DX10)
      THETA2 = ((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10)
      THETA3 = 0.0
      THETA6 = 24.*TAU(L)/(AKW(ISLICE,IS)*S(ISLICE)*(DX1+DX2+DX9+DX10))
      IF (IS.LT.NSTA-1) THETA3 = ((TAU(L)+TAU(LDN))/(2.*DX2))*2.*
     Z      TAU(L)/(DX1+DX2+DX9+DX10)
      THETA4 = 0.0
      THETA5 = 0.0
      HUB1 = 0.0
      HUB3 = 0.0
      TIP1 = 0.0
      TIP3 = 0.0
      IF (ISLICE.EQ.1) GO TO 3410
C
C  FOR A SLICE THAT IS NOT AT THE HUB OF THE BLADE:
C
      THETA4 = (TAU(L)/S(ISLICE))*(TAU(L)/(S(ISLICE)+S(ISLICE-1)))*
     Z           2.*(DX1+DX2)/(DX1+DX2+DX9+DX10)
      IF (ISLICE.EQ.NSLICE) GO TO 3412
      THETA5 = THETA4*(S(ISLICE)+S(ISLICE-1))/(S(ISLICE)+S(ISLICE+1))
      TBELOW = T(1,ISLICE-1,L)
      TABOVE = T(1,ISLICE+1,L)
      GO TO 3414
C
C  FOR THE SLICE AT THE HUB END OF THE BLADE:
C
3410  CONTINUE
      IF (IHUB.EQ.1) HUB1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(2.*(TAU(L)/
     Z                                                     S(1))**2)
C  FOR IHUB = 1, HUB TEMPERATURE IS SPECIFIED**************************
C
      THETA5 = 0.0
      IF (NSLICE.GT.1) THETA5 = (TAU(L)/S(1))*(TAU(L)/(S(1)+S(2)))*
     Z                            (2.*(DX1+DX2)/(DX1+DX2+DX9+DX10))
C
      IF (IHUB.EQ.3) HUB3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/
     Z                                         (AKW(1,IS)*12.*S(1))
C IHUB = 3, THE HEAT FLUX AT THE HUB END IS SPECIFIED (BTU/HR FT**2) ***
C
      TBELOW = T(1,1,L)
      IF (IHUB.EQ.1) TBELOW = THUB(IS)
C
      TABOVE = T(1,1,L)
      IF (NSLICE.GT.1) TABOVE = T(1,2,L)
C
      IF (NSLICE.GT.1) GO TO 3414
C
C  FOR THE SLICE AT THE BLADE TIP, (IF THERE ARE MORE THAN 1 SLICES
C    BEING CONSIDERED) : ********************************
C
3412  CONTINUE
      IF (ITIP.EQ.1) TIP1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(2.*(TAU(L)/
     Z                                              S(NSLICE))**2)
      IF (NSLICE.GT.1) TBELOW = T(1,ISLICE-1,L)
```

```
      IF (ITIP.EQ.3) TIP3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/
     Z                                 (AKW(NSLICE,IS)*12.*S(NSLICE))
      TABOVE = T(1,ISLICE,L)
      IF (ITIP.EQ.1) TABOVE = TTIP(IS)
C
3414  CONTINUE
      THETA9 = 0.0
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) THETA9 = 2.*3600.*RHOM*SPHTM*
     Z     (DX1+DX2)*(TAU(L)**2)/(144.*AKW(ISLICE,IS)*
     Z                  (DX1+DX2+DX9+DX10)*DLTYME)
C
C
C  ENDEFF IS EFFECT OF HEAT TRANSFER FROM THE GAS TO THE REAR EDGE OF
C     THE BLADE
C
      ENDEFF = 0.0
      ENDFLX = 0.0
      IF (IS.GE.NSTA-1.AND.BTA.EQ.0.0) ENDEFF = 2.*HG(IS)*(TAU(L)**2)/
     Z     (12.*AKW(ISLICE,IS)*(DX1+DX9))
      IF (IS.GE.NSTA-1.AND.BTA.GT.0.0) ENDFLX = QG(IS)*(TAU(L)**2)/
     Z     (12.*AKW(ISLICE,IS)*(DX1+DX9))
3416  CONTINUE
      TCOF(L,11) = 1.0*TREPS
      TCOF(L,13) = THETA1*TREPS
      TCOF(L,J1) = THETA2*TREPS
      TCOF(L,J2) = 0.0
      TCOF(L,12) = (-1.0 - THETA1 - THETA2 - THETA3 - THETA4 - THETA5
     Z                -ENDEFF - HUB1 - TIP1)*TREPS - THETA9
      TCOF(L,24) = QSNK(IS)*THETA6 - (THETA4+HUB1)*TBELOW -
     Z     (THETA5+TIP1)*TABOVE - QHUB(IS)*HUB3 + QTIP(IS)*TIP3
     Z          -(1.-TREPS)*(T(1,ISLICE,LUP)*THETA2+T(1,ISLICE,LJ) +
     Z     T(1,ISLICE,LIN)*THETA1+T(1,ISLICE,LDN)*THETA3)
     Z          +T(1,ISLICE,L)*((1.0-TREPS)*(1.+THETA1+THETA2+THETA3+
     Z     THETA4+THETA5+HUB1+TIP1+ENDEFF)
     Z          - THETA9) - TG(IS)*ENDEFF - ENDFLX
      IF (IS.LT.NSTA-1) TCOF(L,J2) = THETA3*TREPS
C
      PUMP(IS) = (.1047198*WS)**2*RR(IS)*(RR(IS)-RR(IS-2))
C
C FOR INNER SURFACE NODE:
C
342   THETA8 = 2.*HX*HC(IS)*AHTRN1*TAU(L)/(12.*AKW(ISLICE,IS)*
     Z                  S(ISLICE)*(DX1+DX2+DX5+DX6))
C
      TCOF(LIN,11) = TREPS
      TCOF(LIN,12) = (-1.0 - THETA8)*TREPS
      TCOF(LIN,I3) = THETA8*(YCONV + YFINS)*TREPS
      TCOF(LIN,24) = - YIMP*THETA8*TOG
     Z             - (1.-TREPS)*(T(1,ISLICE,L)-T(1,ISLICE,LIN)*(1.+THETA8)
     Z              +THETA8*(YCONV+YFINS)*T(1,ISLICE,LCOOL))
C
C  IF THIS IS A TRAILING EDGE, PRESSURE SIDE, STATION, COOLANT NODE
C     COINCIDES WITH SUCTION SIDE COOLANT NODE.
      IF (ISENS.EQ.0) GO TO 343
      TCOF(LCOOL,7) = -1.0
      TCOF(LCOOL,12) = 1.0
      TCOF(LCOOL,24) = 0.0
      IF (ISENS.EQ.1) GO TO 430
343   CONTINUE
C
```

```
C FOR COOLANT NODE:
C  FOR THE SPECIAL CASE OF IS = NFWD+1, GO TO 350
C
      FILMW = WFC(IS)
      IF (IS.GT.NFWD) FILMW = FILMW + WFC(IS+1)
      IF (IS.EQ.NFWD+1) GO TO 350
C
C
      WXCP = WCROS(2,ISLICE,IS)*144.*CPC(IS)*3600.
      IF (IS.EQ.ICOMP.OR.IS.EQ.ICOMS) WXCP = WJ(ISLICE,JS)*144.*
     Z                                  CPC(IS)*3600./2.
C
C
C  DEFINE A COOLANT SIDE TRANSIENT TERM, TRTRMG
C
      TRTRMG = 0.0
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMG = (1.+CPC(IS-2)/CPC(IS))*
     Z                              (P(1,ISLICE,IS-2)/T(1,ISLICE,LCUP)
     Z                           - P(1,ISLICE,IS)/T(1,ISLICE,LCOOL))
     Z        *(A(LCUP)+A(LCOOL))*DLX(LCOOL)/(16.*DLTYME*R*
     Z                                 WCROS(2,ISLICE,IS)*12.)
      TCOF(LCOOL,J4) = HX*TREPS*HC(ISUP)*A1/WXCP
      TCOF(LCOOL,J5) = 144.*3600.*TREPS*WCROS(2,ISLICE,ISUP)*
     Z               (1.+AM2(ISUP)*(GAMC(ISUP)-1.)/2.)*CPC(ISUP)/WXCP
     Z        - HX*TREPS*HC(ISUP)*(YCONVU+YFINSU)*(A1+A3)/WXCP - TRTRMG
      TCOF(LCOOL,11) = HX*TREPS*HC(IS)*A2/WXCP
      TCOF(LCOOL,12) = TREPS*((-FILMW*144.*3600.*CPC(IS)/WXCP)-1.0
     Z                            -((GAMC(IS)-1.)/2.)*AM2(IS)
     Z     - HX*HC(IS)*(A2 + A4)*(YCONV+YFINS)/WXCP) - TRTRMG
      TREDGE = 0.0
      IF (IS.GT.NFWD) TREDGE=((1.-TREPS)*HX/WXCP)*(T(1,ISLICE,LCUPP)*
     Z                     HC(ISUP)*A3 + T(1,ISLICE,LCOOLP)*HC(IS)*A4)
      TCOF(LCOOL,24) = -(CPO*FACTOR*WJ(ISLICE,ISUP)*144.*3600./WXCP)*TOG
     Z - PUMP(IS)/(CPC(IS)*778.*144.*32.2)
     Z     + TOG*(HX*HC(ISUP)*A1*YIMPU + HX*HC(IS)*A2*YIMP)/WXCP
     Z  - T(1,ISLICE,LCUPS)*HX*(1-TREPS)*HC(ISUP)*A1/WXCP
     Z - T(1,ISLICE,LCUP)*(144.*3600.*(1.-TREPS)*WCROS(2,ISLICE,ISUP)
     Z      *(1.+AM2(ISUP)*(GAMC(ISUP)-1.)/2.)*CPC(ISUP)/
     Z                        WXCP - HX*(1.-TREPS)*HC(ISUP)*(YCONVU+
     Z      YFINSU)*(A1+A3)/WXCP - TRTRMG)
     Z  - T(1,ISLICE,LIN)*HX*(1.-TREPS)*HC(IS)*A2/WXCP
     Z  - T(1,ISLICE,LCOOL)*((1.-TREPS)*((-FILMW*144.*3600.*CPC(IS)/
     Z      WXCP)-1.0-((GAMC(IS)-1.)/2.)*AM2(IS)
     Z     - HX*HC(IS)*(A2 + A4)*(YCONV+YFINS)/WXCP) - TRTRMG) - TREDGE
      IF (IS.GT.NFWD) TCOF(LCOOL,J6) = TREPS*HX*HC(ISUP)*A3/WXCP
      IF (IS.GT.NFWD) TCOF(LCOOL,J8) = TREPS*HX*HC(IS)*A4/WXCP
C
C*** OF THE TERMS  YIMP,YFINS,YCONV,  ONLY ONE CAN BE NON-ZERO AT A TIME
C   YIMP  = 1.0 MEANS THAT IMPINGEMENT HEAT TRANSFER IS BEING CONSIDERED
C   YFINS = 1.0 MEANS THAT A PIN FINNED SURFACE IS BEING USED
C   YCONV = 1.0 MEANS A FORCED CONVECTION CORRELATION IS BEING USED
C
C
      IF (TYME.GT.0.) RCHRD = (144./3600.)*AKW(ISLICE,IS)*DLTYME/
     Z                        (RHOM*SPHTM*(((DX1+DX2)/2.)**2))
      IF (TYME.GT.0.) RTRNV = (144./3600.)*AKW(ISLICE,IS)*DLTYME/
     Z                        (RHOM*SPHTM*(TAU(L)**2))
      IF (RCHRD.GT.RCHRDM) RCHRDM = RCHRD
      IF (RTRNV.GT.RTRNVM) RTRNVM = RTRNV
      GO TO 430
```

```
350   CONTINUE                                                         NTARAYT 4861
C                                                                      NTARAYT 4862
C  FOR MIXING ZONE; STATION NO. NFWD+1:                                NTARAYT 4863
C                                                                      NTARAYT 4864
      IF (IHC(IS).EQ.3) CALL HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINSNTARAYT 4865
     Z                                                 ,EFAREA)        NTARAYT 4866
      AHTRN1 = EFAREA(IS)                                              NTARAYT 4867
      IF (IHC(IS).EQ.3)   A2 = EFAREA(IS)/2.                           NTARAYT 4868
      A3 = SPAN*DLX(LCOOLP)/2.                                         NTARAYT 4869
      IF (IHC(IS-1).EQ.3) A3 = EFAREA(IS-1)/2.                         NTARAYT 4870
      A4 = SPAN*DLX(LCOOLP)/2.                                         NTARAYT 4871
      IF (IHC(IS).EQ.3)   A4 = EFAREA(IS+1)/2.                         NTARAYT 4872
      IF (IHC(IS).EQ.3) GO TO 360                                      NTARAYT 4873
C                                                                      NTARAYT 4874
      IF (IHC(IS).EQ.2) HC(IS) = HCFRCD(IS,LCOOL,LIN)                  NTARAYT 4875
      AHTRN1 = (DX5+DX6)*S(ISLICE)/2.                                  NTARAYT 4876
360   CONTINUE                                                         NTARAYT 4877
C                                                                      NTARAYT 4878
      IF (IHC(IS-1).EQ.1) YIMPUU = 1.0                                 NTARAYT 4879
      IF (IHC(IS-1).EQ.2) YCNVUU=1.0+RCVRY*AM2(IS-1)*(GAMC(IS-1)-1.)/2. NTARAYT 4880
      IF (IHC(IS-1).EQ.3) YFNSUU = 1.0                                 NTARAYT 4881
C                                                                      NTARAYT 4882
      WXCP = WCROS(2,ISLICE,IS)*144.*CPC(IS)*3600.                     NTARAYT 4883
      RHOBAR = ((P(1,ISLICE,IS-2)+P(1,ISLICE,IS))/24.)*(1./R)*2.*      NTARAYT 4884
     Z     WCROS(2,ISLICE,IS)/                                         NTARAYT 4885
     Z    (T(1,ISLICE,LCUPP+1)*WCROS(2,ISLICE,IS-1) + T(1,ISLICE,LCUP)* NTARAYT 4886
     Z     WCROS(2,ISLICE,IS-2)                                        NTARAYT 4887
     Z                + TOG*WDUMP + T(1,ISLICE,LCOOL)*WCROS(2,ISLICE,IS)) NTARAYT 4888
      VOLBAR = (A(LCUP)+A(LCUPP+1)+A(LCOOL))*(DLX(LCOOLP)+DLX(LIN))/4.  NTARAYT 4889
      TRTRMJ = 0.0                                                     NTARAYT 4890
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMJ = RHOBAR*VOLBAR/        NTARAYT 4891
     Z                          (2.*DLTYME*(WCROS(2,ISLICE,IS)**2))    NTARAYT 4892
      TCOF(LCOOL,1)   = TREPS*HX*HC(IS-2)*A1/WXCP                      NTARAYT 4893
      TCOF(LCOOL,2)   = TREPS*((WCROS(2,ISLICE,ISUP)/WCROS(2,ISLICE,IS)) NTARAYT 4894
     Z     *(1.+AM2(ISUP)*(GAMC(ISUP)-1.)/2.)*(CPC(ISUP)/              NTARAYT 4895
     Z                CPC(IS)) - HX*HC(ISUP)*(YCONVU+YFINSU)*(A1)/WXCP) NTARAYT 4896
     Z          - TRTRMJ*WCROS(2,ISLICE,IS-2)                          NTARAYT 4897
      TCOF(LCOOL,6)   = TREPS*HX*HC(IS-1)*A3/WXCP                      NTARAYT 4898
      TCOF(LCOOL,7)   = TREPS*((WCROS(2,ISLICE,IS-1)/WCROS(2,ISLICE,IS))*NTARAYT 4899
     Z         (1.+AM2(IS-1)*(GAMC(IS-1)-1.)/2.)*(CPC(IS-1)/           NTARAYT 4900
     Z                CPC(IS)) - HX*HC(IS-1)*(YCNVUU+YFNSUU)*(A3)/WXCP) NTARAYT 4901
     Z        - TRTRMJ*WCROS(2,ISLICE,IS-1)                            NTARAYT 4902
      TCOF(LCOOL,11)  = TREPS*HX*HC(IS)*A2/WXCP                        NTARAYT 4903
      TCOF(LCOOL,12) =TREPS*((-FILMW/WCROS(2,ISLICE,IS))-1.0-((GAMC(IS) NTARAYT 4904
     Z              -1.)/2.)*AM2(IS)-HX*HC(IS)*(A2+A4)*(YCONV+YFINS)   NTARAYT 4905
     Z             /WXCP) - TRTRMJ*WCROS(2,ISLICE,IS)                  NTARAYT 4906
      TCOF(LCOOL,16)  = TREPS*HX*HC(IS)*A4/WXCP                        NTARAYT 4907
      TCOF(LCOOL,24)  = TOG*(HX*HC(IS-2)*A1*YIMPU + HX*HC(IS)*(A2+A4)* NTARAYT 4908
     Z     YIMP + HX*HC(IS-1)*A3*YIMPUU)/WXCP                          NTARAYT 4909
     Z  -PUMP(IS)/(CPC(IS)*778.*144.*32.2) - (CPO/CPC(IS))*TOG*         NTARAYT 4910
     Z     (WJ(ISLICE,IS-2)+WJ(ISLICE,IS-1)+WDUMP)/WCROS(2,ISLICE,IS)  NTARAYT 4911
     Z  -T(1,ISLICE,LCUPS)*(1.-TREPS)*HX*HC(IS-2)*A1/WXCP              NTARAYT 4912
     Z  -T(1,ISLICE,LCUP)*((1.-TREPS)*((WCROS(2,ISLICE,ISUP)/          NTARAYT 4913
     Z     WCROS(2,ISLICE,IS))*(1.+AM2(ISUP)*(GAMC(ISUP)-1.)/2.)       NTARAYT 4914
     Z            *(CPC(ISUP)/CPC(IS)) - HX*HC(ISUP)*(YCONVU+YFINSU)   NTARAYT 4915
     Z     *(A1)/WXCP) + TRTRMJ*WCROS(2,ISLICE,IS-2))                  NTARAYT 4916
     Z  -T(1,ISLICE,LCUPP)*(1.-TREPS)*HX*HC(IS-1)*A3/WXCP              NTARAYT 4917
     Z  -T(1,ISLICE,LCUPP+1)*((1.-TREPS)*((WCROS(2,ISLICE,IS-1)/       NTARAYT 4918
     Z     WCROS(2,ISLICE,IS))*(1.+AM2(IS-1)*(GAMC(IS-1)-1.)/2.)       NTARAYT 4919
     Z        *(CPC(IS-1)/CPC(IS)) - HX*HC(IS-1)*(YCNVUU+YFNSUU)*(A3)/ NTARAYT 4920
```

```
     Z      WXCP) + TRTRMJ*WCROS(2,ISLICE,IS-1))
     Z  -T(1,ISLICE,LIN)*(1.-TREPS)*HX*HC(IS)*A2/WXCP
     Z  -T(1,ISLICE,LCOOL)*(1.-TREPS)*((-FILMW/WCROS(2,ISLICE,IS))-1.0
     Z     -((GAMC(IS)-1.)/2.)*AM2(IS)
     Z                         - HX*HC(IS)*(A2+A4)*(YCONV+YFINS)/WXCP)
     Z  - T(1,ISLICE,LCOOL)*TRTRMJ*WCROS(2,ISLICE,IS)
     Z  -T(1,ISLICE,LCOOLP)*(1.-TREPS)*HX*HC(IS)*A4/WXCP
C
      GO TO 430
C
C
C
C
C
C
C
C
C
C***********************************************************************
C********
C******** THIS BLOCK SETS UP TCOF ELEMENTS FOR STATIONS THAT ARE FORWARD
C            OF THE FLOW SPLIT POINT,
C******** AND ARE ON THE SAME SIDE OF THE BLADE AS THE FLOW SPLIT POINT
C********
C***********************************************************************
C
370   CONTINUE
      ISUP = IS + 2
      LCUP = LCOOL + 10
      LCUPS = LCOOL + 9
      LUP = LCOOL + 8
      LDN = LCOOL - 12
      IF (IS.EQ.2) LDN = 3
      IF (IS.EQ.1) LDN = 8
      IF (IHC(ISUP).EQ.1) YIMPU = 1.0
      IF (IHC(ISUP).EQ.2) YCONVU=1.0+RCVRY*AM2(ISUP)*(GAMC(ISUP)-1.)/2.
      IF (IHC(ISUP).EQ.3) YFINSU = 1.0
      DX1  = DLX(LUP)
      DX2  = DLX(L)
      DX3  = DLX(LUP-2)
      DX4  = DLX(LOUT)
      DX5  = DLX(LCUPS)
      DX6  = DLX(LIN)
      DX7  = DLX(LCUP)
      DX9  = DLX(LUP-1)
      DX10 = DLX(LJ)
      IF (IS.GT.1) GO TO 332
C
      DX2  = DLX(8)
      DX4  = DLX(6)
      DX6  = DLX(9)
      DX10 = DLX(7)
      J1 = 22
      J2 = 17
      J4 = 21
      J5 = 22
      J6 =  8
      J8 = 15
      GO TO 333
C
```

```
C
C
C
C
C
C
C
C*************************************************************************
C********
C******** SETUP FOR TRAILING EDGE REGION:
C********
C*************************************************************************
380   CONTINUE
      ISENS = IS -2*(IS/2)
C
C  ISENS = 0 MEANS IS IS EVEN AND STATION IS ON SUCTION SIDE
C  ISENS = 1 MEANS IS IS ODD AND STATION IS ON PRESSURE SIDE
C
      LCOOL  = 5*IS
      LIN    = LCOOL - 1
      L      = LCOOL - 2
      LJ     = LCOOL - 3
      LOUT   = LCOOL - 4
      LUP    = L - 10
      LDN    = L + 10
      LCUP   = LCOOL - 10
      LCUPS  = LCUP - 1
      LCUPP  = LCOOL - 6
      LCOOLP = LCOOL + 4
C
C
      I3 = 12 - LIN + LCOOL
      J1 = 12 - L + LUP
      J2 = 12 - L + LDN
      J4 = 12 - LCOOL + LCUPS
      J5 = 12 - LCOOL + LCUP
      J6 = 12 - LCOOL + LCUPP
      J8 = 12 - LCOOL + LCOOLP
      J9 = 16
C
      A1 = SPAN*DLX(LIN)/2.
      A2 = A1
      A3 = SPAN*DLX(LCOOLP)/2.
      A4 = A3
      IF (IHC(IS-2).EQ.3) A1 = EFAREA(IS-2)/2.
      IF (IHC(IS-2).EQ.3) A3 = EFAREA(IS-1)/2.
      DX1 = DLX(L)
      DX2 = DLX(LDN)
      DX3 = DLX(LOUT)
      DX4 = DLX(LDN-2)
      DX5 = DLX(LIN)
      DX6 = DLX(LDN+1)
      DX7 = DLX(LCOOL)
      DX9 = DLX(LJ)
      DX10= DLX(LDN-1)
C
      IF (IHC(ISUP).EQ.1) YIMPU = 1.0
      IF (IHC(ISUP).EQ.2) YCONVU=1.0+RCVRY*AM2(ISUP)*(GAMC(ISUP)-1.)/2.
      IF (IHC(ISUP).EQ.3) YFINSU = 1.0
C
```

```
      IF (IS.LT.NSTA-1) GO TO 390
C
C  FOR THE LAST STATIONS IN THE TRAILING EDGE:
C
      DX2 = 0.0
      DX4 = 0.0
      DX6 = 0.0
      DX10= 0.0
390   CONTINUE
      IF ( IHC(IS).EQ.3) GO TO 420
      IF ( IHC(IS).EQ.2) GO TO 410
      GO TO 340
C
C*** HCFRCD COMPUTES HC FOR FORCED CONVECTION
C
410   CONTINUE
      TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
      RE(IS) = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU)
C
      HC(IS) = .023*12.*(C/DH(IS))*(RE(IS)**.8)*(PD**.333)
      AHTRN1 = (DX5 + DX6)*SPAN/2.
      GO TO 340
C
C***** SUBROUTINE HCPINS COMPUTES HC FOR A PIN FIN SURFACE OR FOR
C        TURBULENT FORCED CONVECTION CHANNEL FLOW
420   CONTINUE
      IF(ISENS.EQ.0) GO TO 424
      AHTRN1 = EFAREA(IS)
      HC(IS) = HC(IS-1)
      GO TO 340
C
424   CALL HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EFAREA)
      AHTRN1 = EFAREA(IS)
      IF (IS.GE.NSTA-1) AHTRN1 = AHTRN1/2.
      A2 = EFAREA(IS)/2.
      A4 = EFAREA(IS+1)/2.
      GO TO 340
C
C
C
C
430   CONTINUE
440   CONTINUE
450   CONTINUE
C
      RETURN
      END


C----SOURCE.NTCOFTT
      SUBROUTINE TCOEF(IWRITE,WS,NIT,IPLOT,ALPH2)
C
C-    SOURCE.NTCOFTT
C
      DIMENSION POLD(15,80), PSAV(5), X(80),  ALPH2(4), DELTAN(15),
     Z          TTOTC(80),   JSO(15)
      DIMENSION  PEXOLD(15), PIMOLD(15)
      REAL*8 TCOF
      COMMON /MATRX/ TCOF(400,30)
C
      COMMON /PRPS/ CPO,      GAMO,      DP(80),     SP(80),   RE(80),
```

```
     Z               CPC(80),  GAMC(80),  DUMR1(80),  DUMR2(80)
C
C
      COMMON /TCO/ ADUMP,      BTA,        CD,         CP,
     Z             GAM,        PIM,        R,          SPAN,       TOG,
     Z             WDUMP,      WIM,        AKC(15,80), AKW(15,80),
     Z             A(400),     AJET(80),   AM2(80),    CNUM(80),
     Z             DH(80),     DHF(80),    DHJ(80),
     Z             DLX(400),   FF(80),     HC(80),     HG(80),
     Z             P(2,15,80),PEXIT(15),   PUMP(80),   QG(80),
     Z             QSNK(80),   RR(80),     S(15),      T(2,15,400),
     Z             TG(80),     TAU(400),   WFC(80),
     Z             WJ(15,80),  WCROS(2,15,80),         XN(80),
     Z             ICOR,       IFILM,      IHUB,       ITIP,
     Z             ISBLOK,     ISLICE,     NBLKSZ,     NSLICE,
     Z             NFWD,       NSTA,       IHC(80)
C
      COMMON /TRNSNT/ RHOC,       RHOM,       SPHTC,      SPHTM,
     Z                DLTYME,     TYME,       TEPS,       TYMMAX
C
C
C PROGRAM IS SET TO WORK WITH THE FOLLOWING UNITS ON THE VARIABLES:
C PRESSURE, P, IS IN PSIA ******************************************
C FLOW RATES, WC,WJ,&WCROS ARE IN LBM/SEC **************************
C AREAS ARE IN IN**2 ***********************************************
C ALL LENGTHS ARE IN INCHES ****************************************
C HEAT TRANSFER COEFFICIENTS IN BTU/(HR*FT**2*R) *******************
C
C  BTA = 0. INDICATES THAT A HEAT TRANSFER COEFFICIENT BOUNDARY
C          CONDITION IS SPECIFIED
C        1. INDICATES THAT A HEAT FLUX IS SPECIFIED ON THE GAS BOUNDARY
C  UNITS EXPECTED ON THE FOLLOWING INPUT DATA ARE:
C   LENGTHS ARE ALL IN INCHES-- DLX, TAU, SPAN, DH, DPX
C   TEMPERATURES ARE ABSOLUTE (R)
C   MASS FLOWS ARE ALL IN (LBM/SEC),--- WCROS, WJ, WFC, WDUMP
C   HEAT TRANSFER COEFFICIENTS ARE IN  BTU/(HR*FT**2*R),--HG,HC--GAS
C    SIDE AND COOLANT SIDE .
C   HEAT FLUX QG IS IN BTU/(HR*FT**2)
C   HEAT SINK, QSNK, IS IN BTU/(HR)
C   THERMAL PROPERTIES ARE: CONDUCTIVITY-AKW-BTU/(HR*FT*R)
C                           HEAT CAPACITY-CP-BTU/(LBM*R)
C
C
C----SET UP FIRST GUESS AT TEMPERATURE DISTRIBUTION
C----ASSUME COOLANT TEMPERATURE IS CONSTANT, = PLENUM STATIC TEMPERATURE
C----ASSUME METAL TEMPERATURE IS CONSTANT, = 2200. R
C
100   CONTINUE
      NODST = 5*NSTA
      N = NSTA - 1
      V = .70
      NSAVE = NODST-10
      TSAVE = T(2,ISLICE,NSAVE)
C
C  NODST = NODE NUMBER OF LAST FLOW CHANNEL NODE,AT EXIT OF TRAILING EDG
C
110   TMP = TOG
      NODSF = 5*NFWD
120   CONTINUE
C
```

```
C-- FOR TRANSIENT CASES, ADJUST INITIAL GUESS OF PRESSURE DISTRIBUTION
C-- BASED ON THE VARIATION OF SUPPLY AND EXIT PRESSURES.
C
      IF (TYME.GT.0.0) GO TO 234
C
C-- FOR STEADY STATE, ONLY INITIALIZE P'S ON FIRST OVERALL LOOP.
C
      IF (NIT.GT.1) GO TO 255
      PEXOLD(ISLICE) = PEXIT(ISLICE)
      PIMOLD(ISLICE) = PIM
160   WRITE(6,165)  (I,T(1,ISLICE,I),I=1,NODST)
165   FORMAT(1H1,' ASSUMED INITIAL TEMPERATURE DISTRIBUTION, (NODE NO.',
     Z          ',T) '/7(' (',I3,',',F8.2,'),'))
170   CONTINUE
C
C----PRESSURE INITIALIZATION, GIVEN PIM (PLENUM STATIC PRESSURE) AND
C----PEXIT (GAS SIDE STATIC PRESSURE AT TRAILING EDGE), FIT PRESSURE TO
C     A
C----CUBIC EQN. OF THE FORM A+B*X**3 , ASSUMING 85% OF THE PRESSURE DROP
C----OCCURS IN THE TRAILING EDGE CHANNEL
C
      X(1) = 0.0
      X(2) = DLX(10)
      DO 180 I = 3,NFWD
      X(I) = DLX(5*I) + X(I-2)
180   CONTINUE
      P(1,ISLICE,1) = PIM - (PIM-PFXIT(ISLICE))*.15
      P(1,ISLICE,N) = PEXIT(ISLICE)
      P(1,ISLICE,N+1) = P(1,ISLICE,N)
      P(1,ISLICE,NFWD-1)=P(1,ISLICE,1)-.2*(P(1,ISLICE,1)-PEXIT(ISLICE))
190   P(1,ISLICE,NFWD) = P(1,ISLICE,NFWD-1)
      DO 200 I = 3,NFWD,2
      P(1,ISLICE,I) = P(1,ISLICE,1) - (P(1,ISLICE,1)-P(1,ISLICE,NFWD))*
     Z                                                (X(I)/X(NFWD))**3
200   P(1,ISLICE,I-1)=P(1,ISLICE,1)-(P(1,ISLICE,1)-P(1,ISLICE,NFWD-1))*
     Z                                           (X(I-1)/X(NFWD-1))**3
      ISTRT = NFWD+1
      IFNL = N
C
C----FOR TRAILING EDGE CHANNEL, X VALUES ARE RELATIVE TO END OF
C       IMPINGEMENT CHANNEL
C
      X(NFWD+1) = (DLX(NODSF) + DLX(NODSF-5))/2.
210   ITEM = NFWD+3
      DO 220 I = ITEM,N,2
      LCOOL = 5*I
      X(I) = X(I-2) + DLX(LCOOL)
220   CONTINUE
      DO 230 I = ISTRT,IFNL,2
      P(1,ISLICE,I) = P(1,ISLICE,NFWD)-(P(1,ISLICE,NFWD)-PEXIT(ISLICE))*
     Z                                                  (X(I)/X(N))**3
230   P(1,ISLICE,I+1) = P(1,ISLICE,I)
      DO 232 I = 1,NSTA
232   POLD(ISLICE,I) = P(1,ISLICE,I)
      DO 233 I = 1,NSTA
233   P(2,ISLICE,I) = P(1,ISLICE,I)
      GO TO 240
C
234   DLTAPC = .84*(PIM-PIMOLD(ISLICE))
      PIMOLD(ISLICE) = PIM
```

```
      DLTAPE = PEXIT(ISLICE)-PEXOLD(ISLICE)                              NTCOFTT 5221
      PEXOLD(ISLICE) = PEXIT(ISLICE)                                     NTCOFTT 5222
      DO 235 I = 1,NFWD                                                  NTCOFTT 5223
  235 P(2,ISLICE,I) = P(2,ISLICE,I) + DLTAPC                             NTCOFTT 5224
      ISTRT = NFWD+1                                                     NTCOFTT 5225
      IFNL = NSTA-1                                                      NTCOFTT 5226
      DO 236 I = ISTRT,IFNL,2                                            NTCOFTT 5227
      P(2,ISLICE,I) = P(2,ISLICE,I) + DLTAPC*(1.0-X(I)/X(IFNL)) +        NTCOFTT 5228
     Z                                            DLTAPE*X(I)/X(IFNL)NTCOFTT 5229
  236 P(2,ISLICE,I+1) = P(2,ISLICE,I)                                    NTCOFTT 5230
      DO 237 I = 1,NSTA                                                  NTCOFTT 5231
  237 POLD(ISLICE,I) = P(2,ISLICE,I)                                     NTCOFTT 5232
      GO TO 255                                                          NTCOFTT 5233
C                                                                        NTCOFTT 5234
  240 WRITE(6,245)                                                       NTCOFTT 5235
  245 FORMAT(/' INITIAL PRESSURE DIST.  (STATION NO.,P) '/)              NTCOFTT 5236
      WRITE(6,250) (I,P(1,ISLICE,I),I=1,ISTRT)                           NTCOFTT 5237
      WRITE(6,250) (I,P(1,ISLICE,I),I=ITEM,N,2)                          NTCOFTT 5238
  250 FORMAT(7('  (',I3,',',F7.2,'),'))                                  NTCOFTT 5239
C                                                                        NTCOFTT 5240
C                                                                        NTCOFTT 5241
C                                                                        NTCOFTT 5242
  255 CONTINUE                                                           NTCOFTT 5243
      DO 260 I=1,4                                                       NTCOFTT 5244
  260 PSAV(I)=0.0                                                        NTCOFTT 5245
C                                                                        NTCOFTT 5246
C----INITIALLY, THE FLOW SPLIT AT THE LEADING EDGE IS ASSUMED            NTCOFTT 5247
C      TO BE 50/50 (DELTA=.5)                                            NTCOFTT 5248
C                                                                        NTCOFTT 5249
C----IDELT COUNTS THE NUMBER OF FLOW SPLIT ITERATIONS. IF NO             NTCOFTT 5250
C      CONVERGENCE, IDELT IS SET NEGATIVE.                               NTCOFTT 5251
C----DELTAN IS THE FRACTION OF FLOW TO THE SUCTION SIDE (EVEN            NTCOFTT 5252
C      NUMBERED STATIONS)                                                NTCOFTT 5253
C----IVERGE COUNTS THE NUMBER OF ITERATIONS AT A GIVEN FLOW SPLIT        NTCOFTT 5254
C                                                                        NTCOFTT 5255
C IFNL = THE NUMBER OF FLOW CHANNEL NODES, USED IN PRESSURE CALCULATIONSNTCOFTT 5256
C                                                                        NTCOFTT 5257
      IF (NIT.EQ.1.AND.TYME.LT.0.0) DELTAN(ISLICE) = .5                  NTCOFTT 5258
  275 CONTINUE                                                           NTCOFTT 5259
      IFNL = NSTA - 3                                                    NTCOFTT 5260
      IVERGE = 0                                                         NTCOFTT 5261
      IDELT = 1                                                          NTCOFTT 5262
      JS = 1                                                             NTCOFTT 5263
      IF (NIT.GT.1) JS = JSO(ISLICE)                                     NTCOFTT 5264
  290 CONTINUE                                                           NTCOFTT 5265
      IVERGE = IVERGE + 1                                                NTCOFTT 5266
  300 CONTINUE                                                           NTCOFTT 5267
C                                                                        NTCOFTT 5268
      JSENS = JS - 2*(JS/2)                                              NTCOFTT 5269
C                                                                        NTCOFTT 5270
C----SUBROUTINE FLOWS COMPUTES JET FLOW RATES, CROSSFLOW RATES, AND      NTCOFTT 5271
C     THE SQUARE OF THE MACH NUMBER                                      NTCOFTT 5272
C                                                                        NTCOFTT 5273
  310 CONTINUE                                                           NTCOFTT 5274
C                                                                        NTCOFTT 5275
      CALL FLOWS(JS,DELTAN,ICHOKE,AMCHOK)                                NTCOFTT 5276
  320 CONTINUE                                                           NTCOFTT 5277
      IF (WJ(ISLICE,JS).LE.0.0) GO TO 370                                NTCOFTT 5278
C                                                                        NTCOFTT 5279
C  SUBROUTINE HCOOL COMPUTES IMPINGEMENT REGION HEAT TRANSFER COEFF'NTS NTCOFTT 5280
```

```
C
      CALL HCOOL(JS)
330   CONTINUE
      CALL THRCON
C SUBROUTINE THRCON EXTRACTS THERMAL CONDUCTIVITIES FROM INPUT
C     TABLES AKCTBL AND AKWTBL.
C
335   CONTINUE
C
C---> CHECK TO MAKE SURE STAGNATION HC IS LESS THAN MAXIMUM PHYSICALLY
C        POSSIBLE VALUE.
C
      IF (JS.EQ.1) ASTG = (DLX(9)+DLX(14))*SPAN/2.0
      IF (JS.GT.1) ASTG = (DLX(5*JS-1)+DLX(5*JS+9))*SPAN/2.0
      HSTGMX = WJ(ISLICE,JS)*CPO*144.*3600./ASTG
      IF (HC(JS).GT.HSTGMX) HC(JS) = HSTGMX
C
C
337   CONTINUE
      CALL TAPRAY(JS,JSENS,DELTAN)
C
C SOLVE THE TCOF ARRAY AND SET NEW TEMPERATURE VALUES:
C
340   CONTINUE
      CALL GAUSS(NODST,23)
      DO 350 I = 1,NODST
      T(2,ISLICE,I) = TCOF(I,24)
      IF (T(2,ISLICE,I).LE.0.0) T(2,ISLICE,I) = TOG
350   CONTINUE
C********************
360   IF (ABS((T(2,ISLICE,NSAVE)-TSAVE)/TSAVE) .GT..05)
     Z                          CALL FLOWS(JS,DELTAN,ICHOKE,AMCHOK)
      TSAVE = T(2,ISLICE,NSAVE)
C********************
      IF (ICHOKE.EQ.0) GO TO 370
      WRITE(8,365) ISLICE,IVERGE,IDELT,NIT,ICHOKE,AMCHOK
365   FORMAT(/10X,'SLICE ',I2,' IS CHOKED FOR IVERGE =',I3,',  IDELT =',
     Z   I3,',  NIT =',I3,',  ICHOKE =',I4,',  M**2 =',F6.3)
370   CONTINUE
C
C COMPUTE NEW PRESSURES
C
      CALL PARRAY(JS,JSENS,ICHOKE)
C
C SOLVE THE TCOF ARRAY AND COMPUTE NEW PRESSURES
C
430   CONTINUE
      CALL GAUSS(IFNL,19)
440   CONTINUE
      DO 460 I = 1,IFNL
450   P(2,ISLICE,I) = TCOF(I,20)
460   CONTINUE
      P(2,ISLICE,IFNL+1) = P(2,ISLICE,IFNL)
      P(2,ISLICE,NSTA-1) = PEXIT(ISLICE)
      P(2,ISLICE,NSTA) = PEXIT(ISLICE)
470   CONTINUE
      IF (IWRITE.EQ.2) CALL WROUT(IDELT,JS,DELTAN,IVERGE)
      IF (IPLOT.EQ.2) CALL PLOTMF(ALPH2)
C
480   CONTINUE
```

```
C
C CHECK OVERALL CONVERGENCE
C     CALCULATIONS ARE REPEATED UNTIL THE PRESSURE AT STATION 1 (NODE 5)
C       HAS STABILIZED FOR
C     FOUR ITERATIONS. THEN WE GO TO CHECK THE FLOW SPLIT.
C
      DO 490 I=1,3
      K=5-I
490   PSAV(K)=PSAV(K-1)
      PSAV(1)=P(2,ISLICE,JS)
C
      DIFO=0.0
      DO 500 I=1,3
      JJ=I+1
      DO 500 K=JJ,4
      DIFN=ABS(PSAV(I)-PSAV(K))
      IF(DIFO.LT.DIFN) DIFO=DIFN
500   CONTINUE
C
510   DIFO=DIFO/(PIM-PEXIT(ISLICE))
      PCNVRG = .01
      IF (NIT.EQ.1.AND.NSLICE.GT.1) PCNVRG = .05
      EPSN = (P(2,ISLICE,NFWD-1)-P(2,ISLICE,NFWD))/(P(2,ISLICE,NFWD-1))
      IF (IDELT.EQ.1.AND.IVERGE.LT.3.AND.TYME.LE.0.0) GO TO 516
      IF (DIFO.LE.PCNVRG.AND.IVERGE.GE.4) GO TO 560
516   CONTINUE
      IF (IVERGE.LT.10) GO TO  520
      IF (ABS(PSAV(1)-PSAV(2)).GT.ABS(PSAV(3)-PSAV(4))) V=1.-(1.-V)/2.
520   CONTINUE
      DO 530 I = 1,NSTA
      P(2,ISLICE,I) = P(2,ISLICE,I) + V*(POLD(ISLICE,I) - P(2,ISLICE,I))
      IF (P(2,ISLICE,I).GE.PIM) P(2,ISLICE,I) = .999*PIM
      POLD(ISLICE,I) = P(2,ISLICE,I)
      TTOTC(I) = T(2,ISLICE,5*I)*(1.+(GAMC(I)-1.)*AM2(I)/2.)
530   CONTINUE
C
540   CONTINUE
      IF (IVERGE.GT.30.OR.V.GT..95)
     Z                   WRITE(8,550) IVERGE,V,(PSAV(I),I=1,4)
550   FORMAT('  *********** CONVERGENCE PROBLEMS **************'/
     Z  '    IVERGE=',I3,';  V= ',F6.4,';  PSAV= ',4(F10.2))
      IF (IVERGE.GT.50) GO TO 590
      GO TO 290
C
C********* ONCE PRESSURE-TEMPERATURE ITERATION HAS CONVERGED, CHECK
C             THE FLOW SPLIT AND ADJUST IF NECESSARY.
C
560   CONTINUE
      IF ( IWRITE.EQ.1 ) CALL WROUT(IDELT,JS,DELTAN,IVERGE)
      IF (ICHOKE.GT.0) WRITE(6,565) ICHOKE,AMCHOK
565   FORMAT(/10X,'MACH NO. > 1 AT STATION ',I4,',    M**2 =',F6.3/)
570   CONTINUE
      EPSN = (P(2,ISLICE,NFWD-1)-P(2,ISLICE,NFWD))/(P(2,ISLICE,NFWD-1))
575   CONTINUE
      IF (TYME.GT.0.0) GO TO 590
      CALL FLSPLT(AJET,EPSN,ISLICE,NODSF,IDELT,JS,DELTAN,ICONV)
580   CONTINUE
      IF (ICONV.EQ.1) CALL WROUT(IDELT,JS,DELTAN,IVERGE)
      IVERGE = 0
      IF ( ICONV.EQ.0 ) GO TO 290
```

```
590    CONTINUE                                                         NTCOFTT 5401
       IF (TYME.GT.0.0) CALL WROUT(IDELT,JS,DELTAN,IVERGE)               NTCOFTT 5402
       JSO(ISLICE) = JS                                                 NTCOFTT 5403
       RETURN                                                           NTCOFTT 5404
       END                                                              NTCOFTT 5405


C----SOURCE.NTHRCNT                                                     NTHRCNT 5406
       SUBROUTINE THRCON                                                NTHRCNT 5407
C                                                                       NTHRCNT 5408
C-    SOURCE.NTHRCNT----                                                NTHRCNT 5409
C                                                                       NTHRCNT 5410
       COMMON /BOUND/ BCXS(400),   BCXP(400),   BCHGS(1000), BCHGP(1000),NTHRCNT 5411
     Z                BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),NTHRCNT 5412
     Z                BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),   NTHRCNT 5413
     Z                QHUBIN(400), QHUB(80),    TTIPIN(400), TTIP(80),   NTHRCNT 5414
     Z                QTIPIN(400), QTIP(80),    RHOVG(400),  PEX(400),   NTHRCNT 5415
     Z                BCTIME(50),  TTIO(50),    PTIO(50),    WPLEN,      NTHRCNT 5416
     Z                WSVST(50),   AKCTBL(20),  AKWTBL(20),  NBCS,  NBCP NTHRCNT 5417
C                                                                       NTHRCNT 5418
       COMMON /TCO/ ADUMP,      BTA,        CD,         CP,             NTHRCNT 5419
     Z              GAM,        FIM,        R,          SPAN,      TOG, NTHRCNT 5420
     Z              WDUMP,      WIM,        AKC(15,80), AKW(15,80),     NTHRCNT 5421
     Z              A(400),     AJET(80),   AM2(80),    CNUM(80),       NTHRCNT 5422
     Z              DH(80),     DHF(80),    DHJ(80),                    NTHRCNT 5423
     Z              DLX(400),   FF(80),     HC(80),     HG(80),         NTHRCNT 5424
     Z              P(2,15,80),PEXIT(15),   PUMP(80),   QG(80),         NTHRCNT 5425
     Z              QSNK(80),   RR(80),     S(15),      T(2,15,400),    NTHRCNT 5426
     Z              TG(80),     TAU(400),   WFC(80),                    NTHRCNT 5427
     Z              WJ(15,80),  WCROS(2,15,80),         XN(80),         NTHRCNT 5428
     Z              ICOR,       IFILM,      IHUB,       ITIP,           NTHRCNT 5429
     Z              ISBLOK,     ISLICE,     NBLKSZ,     NSLICE,         NTHRCNT 5430
     Z              NFWD,       NSTA,       IHC(80)                     NTHRCNT 5431
C                                                                       NTHRCNT 5432
       DO 1000 I = 1,NSTA                                               NTHRCNT 5433
       L = 5*I - 2                                                      NTHRCNT 5434
       LJ = L - 1                                                       NTHRCNT 5435
       LOUT = L - 2                                                     NTHRCNT 5436
       TC = (T(2,ISLICE,LJ) + T(2,ISLICE,LOUT))/2.0 - 460.              NTHRCNT 5437
       TW = T(2,ISLICE,L) - 460.                                        NTHRCNT 5438
C                                                                       NTHRCNT 5439
C  LOOK UP COATING THERMAL CONDUCTIVITY IN TABLE AKCTBL.                NTHRCNT 5440
C                                                                       NTHRCNT 5441
       IF (TC.GT.AKCTBL(1)) GO TO 150                                   NTHRCNT 5442
C                                                                       NTHRCNT 5443
C  FOR A TEMPERATURE LOWER THAN THE BOTTOM OF THE TABLE, EXTRAPOLATE    NTHRCNT 5444
C     BELOW TABLE                                                       NTHRCNT 5445
       RATIO = (TC - AKCTBL(1))/(AKCTBL(3) - AKCTBL(1))                 NTHRCNT 5446
       AKC(ISLICE,I) = AKCTBL(2) + (AKCTBL(4)-AKCTBL(2))*RATIO          NTHRCNT 5447
       GO TO 500                                                        NTHRCNT 5448
C                                                                       NTHRCNT 5449
150    CONTINUE                                                         NTHRCNT 5450
C                                                                       NTHRCNT 5451
C  FIND SIZE OF TABLE                                                   NTHRCNT 5452
C                                                                       NTHRCNT 5453
       DO 152 J = 3,19,2                                                NTHRCNT 5454
       JLST = J-1                                                       NTHRCNT 5455
       IF (AKCTBL(J).LE.0.1) GO TO 154                                  NTHRCNT 5456
152    CONTINUE                                                         NTHRCNT 5457
154    JLSTM = JLST-1                                                   NTHRCNT 5458
C                                                                       NTHRCNT 5459
C  LOCATE WHERE TEMPERATURE FALLS IN THE TABLE AKCTBL.                  NTHRCNT 5460
```

```
C
      DO 170 J = 3,JLSTM,2
      IF (TC.GT.AKCTBL(J)) GO TO 160
C
C  FOUND LOCATION, NOW INTERPOLATE.
C
      RATIO = (TC - AKCTBL(J-2))/(AKCTBL(J) - AKCTBL(J-2))
      AKC(ISLICE,I) = AKCTBL(J-1) + (AKCTBL(J+1) - AKCTBL(J-1))*RATIO
      GO TO 500
160   IF (J.LT.JLSTM) GO TO 170
C
C  TEMPERATURE IS ABOVE THE RANGE OF THE TABLE, SO EXTRAPOLATE UP.
C
      RATIO = (TC - AKCTBL(J-2))/(AKCTBL(J) - AKCTBL(J-2))
      AKC(ISLICE,I) = AKCTBL(J-1) + (AKCTBL(J+1) - AKCTBL(J-1))*RATIO
      GO TO 500
170   CONTINUE
500   CONTINUE
C
C  NOW LOOK UP METAL CONDUCTIVITY IN TABLE AKWTBL.
C
      IF (TW.GT.AKWTBL(1)) GO TO 550
C
C  FOR A TEMPERATURE LOWER THAN THE BOTTOM OF THE TABLE, EXTRAPOLATE
C     BELOW TABLE
      RATIO = (TW - AKWTBL(1))/(AKWTBL(3) - AKWTBL(1))
      AKW(ISLICE,I) = AKWTBL(2) + (AKWTBL(4)-AKWTBL(2))*RATIO
      GO TO 1000
C
550   CONTINUE
C
C  FIND SIZE OF TABLE
C
      DO 552 J = 3,19,2
      JLST = J-1
      IF (AKWTBL(J).LE.0.1) GO TO 554
552   CONTINUE
554   JLSTM = JLST-1
C
C  LOCATE WHERE TEMPERATURE FALLS IN THE TABLE AKWTBL.
C
      DO 570 J = 3,JLSTM,2
      IF (TW.GT.AKWTBL(J)) GO TO 560
C
C  FOUND LOCATION, NOW INTERPOLATE.
C
      RATIO = (TW - AKWTBL(J-2))/(AKWTBL(J) - AKWTBL(J-2))
      AKW(ISLICE,I) = AKWTBL(J-1) + (AKWTBL(J+1) - AKWTBL(J-1))*RATIO
      GO TO 1000
560   IF (J.LT.JLSTM) GO TO 570
C
C  TEMPERATURE IS ABOVE THE RANGE OF THE TABLE, SO EXTRAPOLATE UP.
C
      RATIO = (TW - AKWTBL(J-2))/(AKWTBL(J) - AKWTBL(J-2))
      AKW(ISLICE,I) = AKWTBL(J-1) + (AKWTBL(J+1) - AKWTBL(J-1))*RATIO
      GO TO 1000
570   CONTINUE
1000  CONTINUE
      RETURN
      END
```

```
C----SOURCE.NWROTTT
      SUBROUTINE WROUT(IDELT,JS,DELTAN,IVERGE)
C
C-    SOURCE.NWROTTT----
C
C  DUMR2 CARRIES THE IMPINGEMENT JET REYNOLDS NO. IN FROM HCCOOL.
C
      COMMON /CHKHOL/  WCHK(80),  WCHKDM
C
      COMMON /FLMCOL/ RHOVGA(80), PG(80),   XFC(80),  FLMEFF(80),
     Z                XMUC(80),   EMES(80), REFC(80), NFCSUP(80)
C
      COMMON /PRPS/ CPO,      GAMO,      DP(80),    SP(80),  RE(80),
     Z              CPC(80),  GAMC(80),  DUMR1(80), DUMR2(80)
C
      COMMON /RADL/ APLN(15),  DPLN(15),   RIN(15),   ROUT(15),
     Z              PIN(15),   TIN(15),    W(15),     WS
C
      COMMON /TCO/ ADUMP,     BTA,       CD,         CP,
     Z             GAM,       PIM,       R,          SPAN,       TOG,
     Z             WDUMP,     WIM,       AKC(15,80), AKW(15,80),
     Z             A(400),    AJET(80),  AM2(80),    CNUM(80),
     Z             DH(80),    DHF(80),   DHJ(80),
     Z             DLX(400),  FF(80),    HC(80),     HG(80),
     Z             P(2,15,80),PEXIT(15), PUMP(80),   QG(80),
     Z             QSNK(80),  RR(80),    S(15),      T(2,15,400),
     Z             TG(80),    TAU(400),  WFC(80),
     Z             WJ(15,80), WCROS(2,15,80),        XN(80),
     Z             ICOR,      IFILM,     IHUB,       ITIP,
     Z             ISBLOK,    ISLICE,    NBLKSZ,     NSLICE,
     Z             NFWD,      NSTA,      IHC(80)
C
      COMMON /TRNSNT/ RHOC,      RHOM,      SPHTC,      SPHTM,
     Z                DLTYME,    TYME,      TEPS,       TYMMAX
C
      COMMON /UNITS/ CINCH(2), CHTC(2),  CHFLX(2), CPRSR(2), CMSFL(2),
     Z               CTMPF(2), CTCON(2), CDEN(2),  CSPHT(2), CGASC(2),
     Z               CVISC(2), CRHOVG(2),IUNITS
C
      DIMENSION DUM1(10),DUM2(10),DELTAN(15)
10    CONTINUE
      IF (ISLICE.EQ.1) TBULK = 0.0
      IF (ISLICE.EQ.1) TOTSPN = 0.0
      TTYME = TYME
      IF (TTYME.LT.0.) TTYME=0.0
      WRITE(6,90) TTYME,DLTYME,WS
90    FORMAT(1H1,10X,' TIME = ',F6.2,' SEC.,  STEP SIZE = ',F6.3,
     Z          ' SEC.,   WHEEL SPEED =',F8.1,' RPM')
8805  WRITE(6,8806) ISLICE,IDELT,JS,DELTAN(ISLICE),IVERGE
      ITRBG = NFWD + 2
      WRITE(6,100)  ITRBG
C
      IF (IUNITS.EQ.2) WRITE(6,270)
      IF (IUNITS.EQ.1) WRITE(6,271)
      DO 210 I = 1,NSTA,2
      II = I
      LCOOL = 5*II
      NOS = LCOOL - 4
      DO 205 J = 1,4
      JM = NOS+J-1
```

```
      DUM1(J) = T(2,ISLICE,JM) - 460.
      IF (IUNITS.EQ.1) DUM1(J) = T(2,ISLICE,JM)/CTMPF(1)
  205 CONTINUE
      DUM1(5) = T(2,ISLICE,LCOOL) - 460.
      IF (IUNITS.EQ.1) DUM1(5) = T(2,ISLICE,LCOOL)/CTMPF(1)
      DUM1(6) = P(2,ISLICE,II)/CPRSR(IUNITS)
      DUM1(7) = DUM1(6)*(1.+(GAMC(II)-1.)*AM2(II)/2.)**(GAMC(II)/
     Z                                                  (GAMC(II)-1.))
      DUM1(8) = HC(II)/CHTC(IUNITS)
      DUM1(9) = HG(II)/CHTC(IUNITS)
      DUM1(10)= TG(II) - 460.
      IF (IUNITS.EQ.1) DUM1(10) = TG(II)/1.8
      IF (BTA.GT..001) DUM1(10) = 9.E20
      IF (BTA.GT..001) DUM1(9) = 9.E20
      WRITE(6,274) (II,LCOOL,(DUM1(J),J=1,10))
      IF (I.EQ.NFWD) WRITE(6,276)
  210 CONTINUE
      IF (IUNITS.EQ.2) WRITE(6,278)
      IF (IUNITS.EQ.1) WRITE(6,279)
      DO 220 I = 1,NSTA,2
      II = I
      LCOOL = 5*II
      NOS = LCOOL - 4
      DUM2(1) = WJ(ISLICE,II)/CMSFL(IUNITS)
      DUM2(2) = DUMR2(II)
      DUM2(3) = WCROS(2,ISLICE,II)/CMSFL(IUNITS)
      DUM2(4) = RE(II)
      DUM2(5) = SQRT(AM2(II))
      DUM2(6) = FF(II)
      DUM2(7) = WFC(II)/CMSFL(IUNITS)
      DUM2(8) = FLMEFF(II)
      WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II),(DUM2(J),J=2,8))
      IF (I.EQ.NFWD) WRITE(6,276)
  220 CONTINUE
      DUM2(9) = WDUMP/CMSFL(IUNITS)
      IF (ADUMP.GT.0.0.AND.IUNITS.EQ.2) WRITE(6,290) DUM2(9),WCHKDM
      IF (ADUMP.GT.0.0.AND.IUNITS.EQ.1) WRITE(6,291) DUM2(9),WCHKDM
C
      ITRBG = NFWD + 1
      WRITE(6,124) ISLICE,ITRBG
C
      IF (IUNITS.EQ.2) WRITE(6,270)
      IF (IUNITS.EQ.1) WRITE(6,271)
      DO 230 I = 1,NSTA,2
      II = I
      IF (I.GT.1) II = I-1
      LCOOL = 5*II
      NOS = LCOOL - 4
      DO 225 J = 1,5
      JM = NOS+J-1
      DUM1(J) = T(2,ISLICE,JM) - 460.
      IF (IUNITS.EQ.1) DUM1(J) = T(2,ISLICE,JM)/CTMPF(1)
  225 CONTINUE
      DUM1(6) = P(2,ISLICE,II)/CPRSR(IUNITS)
      DUM1(7) = DUM1(6)*(1.+(GAMC(II)-1.)*AM2(II)/2.)**(GAMC(II)/
     Z                                                  (GAMC(II)-1.))
      DUM1(8) = HC(II)/CHTC(IUNITS)
      DUM1(9) = HG(II)/CHTC(IUNITS)
      DUM1(10)= TG(II) - 460.
      IF (IUNITS.EQ.1) DUM1(10) = TG(II)/1.8
```

```
      IF (BTA.GT..001) DUM1(10) = 9.E20
      IF (BTA.GT..001) DUM1(9) = 9.E20
      WRITE(6,274) (II,LCOOL,(DUM1(J),J=1,10))
      IF (I.EQ.NFWD) WRITE(6,276)
230   CONTINUE
      IF (IUNITS.EQ.2) WRITE(6,278)
      IF (IUNITS.EQ.1) WRITE(6,279)
      DO 240 I = 1,NSTA,2
      II = I
      IF (I.GT.1) II = I-1
      LCOOL = 5*II
      NOS = LCOOL - 4
      DUM2(1) = WJ(ISLICE,II)/CMSFL(IUNITS)
      DUM2(2) = DUMR2(II)
      DUM2(3) = WCROS(2,ISLICE,II)/CMSFL(IUNITS)
      DUM2(4) = RE(II)
      DUM2(5) = SQRT(AM2(II))
      DUM2(6) = FF(II)
      DUM2(7) = WFC(II)/CMSFL(IUNITS)
      DUM2(8) = FLMEFF(II)
      WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II),(DUM2(J),J=2,8))
      IF (I.EQ.NFWD) WRITE(6,276)
240   CONTINUE
      DUM2(9) = WDUMP/CMSFL(IUNITS)
      IF (ADUMP.GT.0.0.AND.IUNITS.EQ.2) WRITE(6,290) DUM2(9),WCHKDM
      IF (ADUMP.GT.0.0.AND.IUNITS.EQ.1) WRITE(6,291) DUM2(9),WCHKDM
C
1000  CONTINUE
C
C   TO DETERMINE THE MEAN OUTSIDE SURFACE TEMPERATURE FOR EACH
C    SIDE OF THE BLADE, AND
C   TO LOCATE THE EXTREME TEMPERATURE POINTS, BOTH HIGH AND LOW.
C
      XTOT = 0.
      XTOTMD = 0.0
      HBAR = HC(1)*(DLX(2)+DLX(3))/2.
      TBAR = T(2,ISLICE,1)*(DLX(6)+DLX(11))/2.
      TBARMD = T(2,ISLICE,3)*(DLX(8)+DLX(13))/2.
      ISTAT = 1
      DO 1004 I = 2,NSTA
      NODM = 5*I-2
      ISTAT = ISTAT + 5
      ISTATD = ISTAT + 10
      TBARMD = TBARMD + T(2,ISLICE,NODM)*(DLX(NODM)+DLX(NODM+10))/2.
      XTOTMD = XTOTMD + DLX(NODM)
      TBAR = TBAR + T(2,ISLICE,ISTAT)*(DLX(ISTAT)+DLX(ISTATD))/2.
      XTOT = XTOT + DLX(ISTAT)
      HBAR = HBAR + HC(I)*(DLX(ISTAT)+DLX(ISTATD))/2.
1004  CONTINUE
      IF (IUNITS.EQ.1) TBAR = TBAR/(1.8*XTOT)
      IF (IUNITS.EQ.2) TBAR = TBAR/XTOT - 460.
      IF (IUNITS.EQ.1) TBARMD = TBARMD/(1.8*XTOTMD)
      IF (IUNITS.EQ.2) TBARMD = TBARMD/XTOTMD -460.
      TBULK = TBULK + TBARMD*S(ISLICE)
      TOTSPN = TOTSPN + S(ISLICE)
      HBAR = HBAR/(XTOT*CHTC(IUNITS))
1008  IF (IUNITS.EQ.2) WRITE(6,1115) TBAR,TBARMD,HBAR
      IF (IUNITS.EQ.1) WRITE(6,1116) TBAR,TBARMD,HBAR
C
      TSMAX  = T(2,ISLICE,1) - 460.
```

```
      TSMIN  = T(2,ISLICE,1) - 460.
      TPMAX  = T(2,ISLICE,1) - 460.
      TPMIN  = T(2,ISLICE,1) - 460.
      ISUCMX = 1
      ISUCMN = 1
      IPRSMX = 1
      IPRSMN = 1
      IPRES  = 1
      ISUCT  = -4
C
      DO 1080 I = 3,NSTA,2
      IPRES  = IPRES + 10
      ISUCT  = ISUCT + 10
C
C
      IF (T(2,ISLICE,IPRES)-460..GT.TPMAX) GO TO 1030
      IF (T(2,ISLICE,IPRES)-460..LT.TPMIN) GO TO 1040
      GO TO 1050
C
1030  TPMAX  = T(2,ISLICE,IPRES)-460.
      IPRSMX = I
      GO TO 1050
C
1040  TPMIN  = T(2,ISLICE,IPRES)-460.
      IPRSMN = I
C
1050  IF (T(2,ISLICE,ISUCT)-460..GT.TSMAX) GO TO 1060
      IF (T(2,ISLICE,ISUCT)-460..LT.TSMIN) GO TO 1070
      GO TO 1080
C
1060  TSMAX  = T(2,ISLICE,ISUCT)-460.
      ISUCMX = I - 1
      GO TO 1080
C
1070  TSMIN  = T(2,ISLICE,ISUCT)-460.
      ISUCMN = I - 1
1080  CONTINUE
C
C
      IF (ISLICE.LT.NSLICE) GO TO 1095
      TBULK = TBULK/TOTSPN
      IF (IUNITS.EQ.1) WRITE(6,1091) TYME,TBULK
      IF (IUNITS.EQ.2) WRITE(6,1090) TYME,TBULK
C
1090  FORMAT(1H2,30X,'TIME =',F6.3,' SEC.,  OVERALL BULK TEMPERATURE =',
     Z                                    F7.1,' F')
1091  FORMAT(1H2,30X,'TIME =',F6.3,' SEC.,  OVERALL BULK TEMPERATURE =',
     Z                                    F7.1,' K')
1095  CONTINUE
1115  FORMAT(/' OVERALL AREA WEIGHTED AVERAGES--OUTSIDE T =',F7.1,
     Z       ' F, MID-WALL T =',F7.1,
     Z       ' F, COOLANT H = 'F6.1,' BTU/(HR/FT**2/R)')
1116  FORMAT(/' OVERALL AREA WEIGHTED AVERAGES--OUTSIDE T =',F7.1,
     Z       ' K, MID-WALL T =',F7.1,
     Z       ' K, COOLANT H = 'F6.1,' WATTS/M**2/K')
      IF (IUNITS.EQ.2) WRITE(6,1120) TPMAX,IPRSMX,TPMIN,IPRSMN,TSMAX,
     Z                 ISUCMX,TSMIN,ISUCMN
1120  FORMAT( /12X,'EXTREMES OF OUTER SURFACE TEMPERATURES (F)'/6X,
     Z      'PRESSURE SIDE: ',
     Z       F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2/6X,
```

```
     Z      'SUCTION  SIDE: ',                                          NWROTTT 5761
     Z        F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2)        NWROTTT 5762
      IF (IUNITS.EQ.2) GO TO 1130                                        NWROTTT 5763
      TPMAX = (TPMAX+460.)/1.8                                           NWROTTT 5764
      TPMIN = (TPMIN+460.)/1.8                                           NWROTTT 5765
      TSMAX = (TSMAX+460.)/1.8                                           NWROTTT 5766
      TSMIN = (TSMIN+460.)/1.8                                           NWROTTT 5767
      WRITE(6,1125) TPMAX,IPRSMX,TPMIN,IPRSMN,TSMAX,ISUCMX,TSMIN,ISUCMN NWROTTT 5768
1125  FORMAT( /12X,'EXTREMES OF OUTER SURFACE TEMPERATURES (K)'/6X,      NWROTTT 5769
     Z      'PRESSURE SIDE: ',                                          NWROTTT 5770
     Z        F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2/6X,     NWROTTT 5771
     Z      'SUCTION  SIDE: ',                                          NWROTTT 5772
     Z        F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2)        NWROTTT 5773
1130  CONTINUE                                                           NWROTTT 5774
100   FORMAT(30X,'PRESSURE SIDE , TRAILING EDGE REGION BEGINS AT ',      NWROTTT 5775
     Z            'STATION-',I3)                                        NWROTTT 5776
124   FORMAT(1H1,/' SLICE NO. ',I2,17X,'SUCTION SIDE , TRAILING EDGE ', NWROTTT 5777
     Z         'REGION BEGINS AT STATION-',I3)                          NWROTTT 5778
8806  FORMAT('   SLICE NO.',I2,', FLOW SPLIT NO.',I3,', SPLIT AT ',      NWROTTT 5779
     Z          'STATION',I3,                                           NWROTTT 5780
     Z  '; FRACTION SPLIT TO SUCTION SIDE IS',F7.4,I6,' ITERATIONS')     NWROTTT 5781
270   FORMAT(/                                                           NWROTTT 5782
     Z ' STATION*COOLANT* OUTSIDE *INTERFACE* MID-WALL* INSIDE  * ',     NWROTTT 5783
     Z  'COOLANT * STATIC P* TOTAL P * HC,BTU/HR* HG,BTU/HR*   TG'       NWROTTT 5784
     Z/' NUMBER *NODE NO*  T (F)   *  T (F)   *  T (F)   *  T (F)   *  T ', NWROTTT 5785
     Z   '(F)   *   (PSIA) *   (PSIA) * /FT**2/R * /FT**2/R *    (F) '    NWROTTT 5786
     Z/119('*')/)                                                        NWROTTT 5787
271   FORMAT(/                                                           NWROTTT 5788
     Z ' STATION*COOLANT* OUTSIDE *INTERFACE* MID-WALL* INSIDE  * ',     NWROTTT 5789
     Z    'COOLANT * STATIC P* TOTAL P *    HC     *     HG     *    TG'  NWROTTT 5790
     Z/' NUMBER *NODE NO*  T (K)   *  T (K)   *  T (K)   *  T (K)   *  T ', NWROTTT 5791
     Z    '(K)   *   (KPA)   *   (KPA)   * W/M**2/K * W/M**2/K *    (K)'   NWROTTT 5792
     Z/119('*')/)                                                        NWROTTT 5793
274   FORMAT(I6,2X,I6,1X,7F10.1,3F11.1)                                  NWROTTT 5794
276   FOPMAT(47X,'BEGIN TRAILING EDGE REGION')                           NWROTTT 5795
278   FORMAT(1H2,/' STATION * COOLANT * IMP. FLOW * RE-NO. * CROSSFLOW',NWROTTT 5796
     Z    ' *   RE-NO.    * MACH NO., *'                                 NWROTTT 5797
     Z       ' FRICTION * FILM FLOW * EFFECTIVENESS *'/                  NWROTTT 5798
     Z              '  NUMBER * NODE NO * (LBM/SEC) *  JET    * (LBM/SEC) ',NWROTTT 5799
     Z    '* CROSSFLOW * CROSSFLOW *'                                    NWROTTT 5800
     Z       '   FACTOR   *  (LBM/SEC) *'/115('*')/19X,'*',20X,'*',46X,'*') NWROTTT 5801
279   FORMAT(1H2,/' STATION * COOLANT * IMP. FLOW * RE-NO. * CROSSFLOW',NWROTTT 5802
     Z    ' *   RE-NO.    * MACH NO., *'                                 NWROTTT 5803
     Z       ' FRICTION * FILM FLOW * EFFECTIVENESS *'/                  NWROTTT 5804
     Z              '  NUMBER * NODE NO *   (KG/SEC) *  JET    *   (KG/SEC) ',NWROTTT 5805
     Z    '* CROSSFLOW * CROSSFLOW *'                                    NWROTTT 5806
     Z       '   FACTOR   *   (KG/SEC) *'/115('*')/19X,'*',20X,'*',46X,'*') NWROTTT 5807
280   FORMAT(1H ,I5,5X,I5,'     *',F9.6,A1,F9.1,' * ',F9.6,2X,F9.1,4X,F9.6NWROTTT 5808
     Z      ,2X,F9.6,' * ',F9.6,4X,F9.6)                                 NWROTTT 5809
290   FORMAT(//15X,' FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION IS 'NWROTTT 5810
     Z      ,F10.6,A1,' LBM/SEC')                                        NWROTTT 5811
291   FORMAT(//15X,' FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION IS 'NWROTTT 5812
     Z      ,F10.6,A1,' KG/SEC')                                         NWROTTT 5813
      RETURN                                                             NWROTTT 5814
      END                                                                NWROTTT 5815
```

REFERENCES

1. Gaugler, Raymond E.: TACT1, A Computer Program for the Transient Thermal Analysis of a Cooled Turbine Blade or Vane Equipped with a Coolant Insert. I - Users Manual. NASA TP-1271, 1978.

2. Poferl, David J.; and Svehla, Roger A.: Thermodynamic and Transport Properties of Air and Its Products of Combustion with ASTM-A-1 Fuel and Natural Gas at 20, 30, and 40 Atmospheres. NASA TN D-7488, 1973.

3. Stollery, J. L.; and El-Ehwany, A. A. M.: A Note on the Use of a Boundary-Layer Model for Correlating Film-Cooling Data. Int. J. Heat Mass Transfer, vol. 8, 1965, pp. 55-65.

TABLE I. - SUBROUTINE CALLS AND COMMON BLOCKS

Subroutine name	Source module	COMMON blocks	Called subroutines	Calling subroutines
BLOCK DATA	NGASDAT	/GAAS/	NONE	NONE
FLOWS	NFLOEST	/CHKHOL/ /FLMCOL/ /FRIC/ /PRPS/ /TCO/ /TRNSNT/	GASTBL	TCOEF
FLSPLT	NFLSPLP	NONE	NONE	TCOEF
GASTBL	NGASTB	/GAAS/	NONE	FLOWS HCFRCD HCOOL HCPINS PLNUM TARRAY
GAUSS	NGAUS	/MATRX/	NONE	TCOEF
GETIN	NGETINT	/BOUND/ /FLMCOL/ /IMPCOR/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS	INPRT	MAIN PROGRAM
HCFRCD	NHCFRCT	/TCO/	GASTBL	HCOOL TARRAY
HCOOL	NHCOOLT	/IMPCOR/ /PRPS/ /TCO/	GASTBL HCFRCD	TCOEF
HCPINS	NHCPINT	/PRPS/ /TCO/	GASTBL	TARRAY
INPRT	NINPRTT	/BOUND/ /GAAS/ /PRPS/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	PREP	GETIN
MAIN PROGRAM	NTTACT	/BOUND/ /FLMCOL/ /GAAS/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	GETIN PLNUM PLOTMF PREP TCOEF	NONE
PARRAY	NPARAYT	/MATRX/ /PRPS/ /TCO/ /TRNSNT/	NONE	TCOEF
PLNUM	NPLENMP	/RADL/ /TCO/ /TRNSNT/ /UNITS/	GASTBL	MAIN PROGRAM
PLOTMF	NPLOTIT	/PRPS/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	NONE	TCOEF MAIN PROGRAM
PREP	NPREPAT	/BOUND/ /FLMCOL/ /FRIC/ /PRPS/ /SPECL/ /TCO/ /TRNSNT	NONE	INPRT MAIN PROGRAM
TARRAY	NTARAYT	/BOUND/ /FLMCOL/ /MATRX/ /PRPS/ /TCO/ /TRNSNT/	GASTBL HCFRCD HCPINS	TCOEF
TCOEF	NTCOFTT	/MATRX/ /PRPS/ /TCO/ /TRNSNT/	FLOWS FLSPLT GAUSS HCOOL PARRAY PLOTMF TARRAY THRCON WROUT	MAIN PROGRAM
THRCON	NTHRCNT	/BOUND/ /TCO/	NONE	TCOEF
WROUT	NWROTTT	/CHKHOL/ /FLMCOL/ /PRPS/ /RADL/ /TCO/ /TRNSNT/ /UNITS/	NONE	TCOEF

TABLE II. - COMMON-BLOCK CROSS-REFERENCE TABLE

Subroutine	COMMON block												
	BOUND	CHKHOL	FLMCOL	FRIC	GAAS	IMPCOR	MATRX	PRPS	RADL	SPECL	TCO	TRNSNT	UNITS
BLOCK DATA					×								
FLOWS		×	×	×				×			×	×	
FLSPLT													
GASTBL					×								
GAUSS							×						
GETIN	×		×			×			×	×	×	×	×
HCFRCD											×		
HCOOL						×		×			×		
HCPINS								×			×		
INPRT	×				×			×	×	×	×	×	×
NTTACT	×		×		×				×	×	×	×	×
PARRAY							×	×			×	×	
PLNUM									×		×	×	×
PLOTMF								×		×	×	×	×
PREP	×		×	×				×		×	×	×	
TARRAY	×		×				×	×			×	×	
TCOEF							×	×			×	×	
THRCON	×										×		
WROUT		×	×					×	×		×	×	×

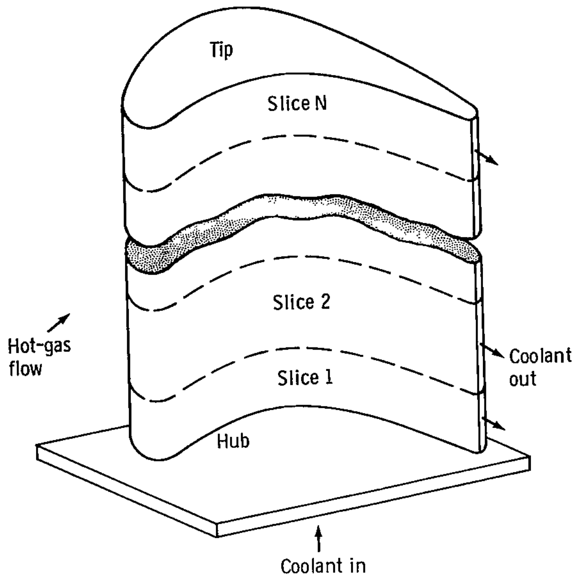


Figure 1. - Overall division of blade into slices.

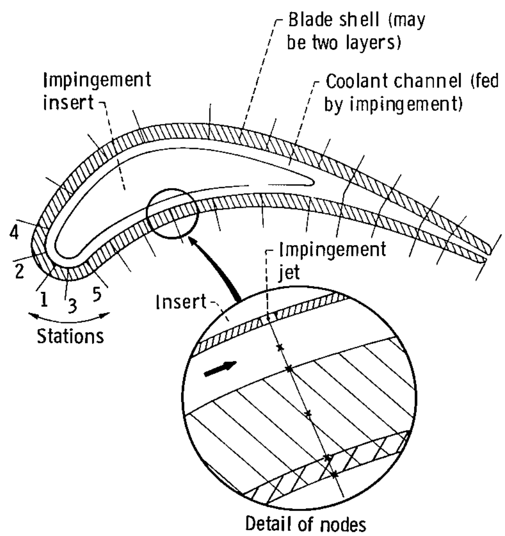


Figure 2. - Blade geometric model.

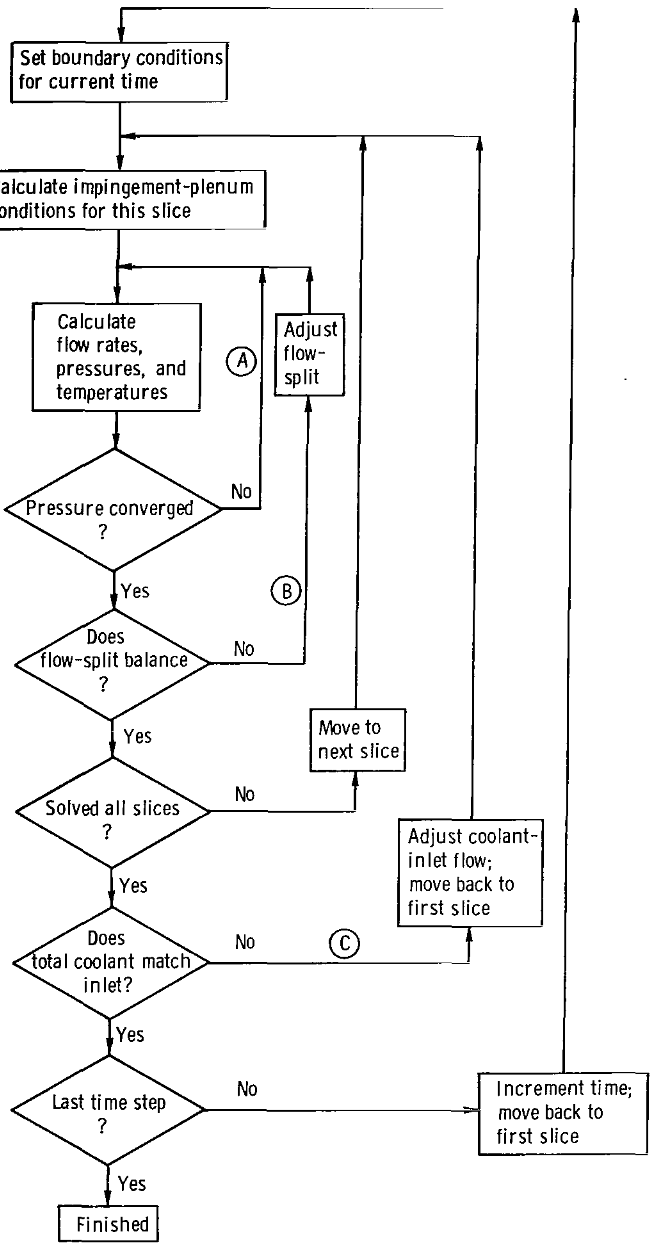


Figure 3. - Overall program procedure.

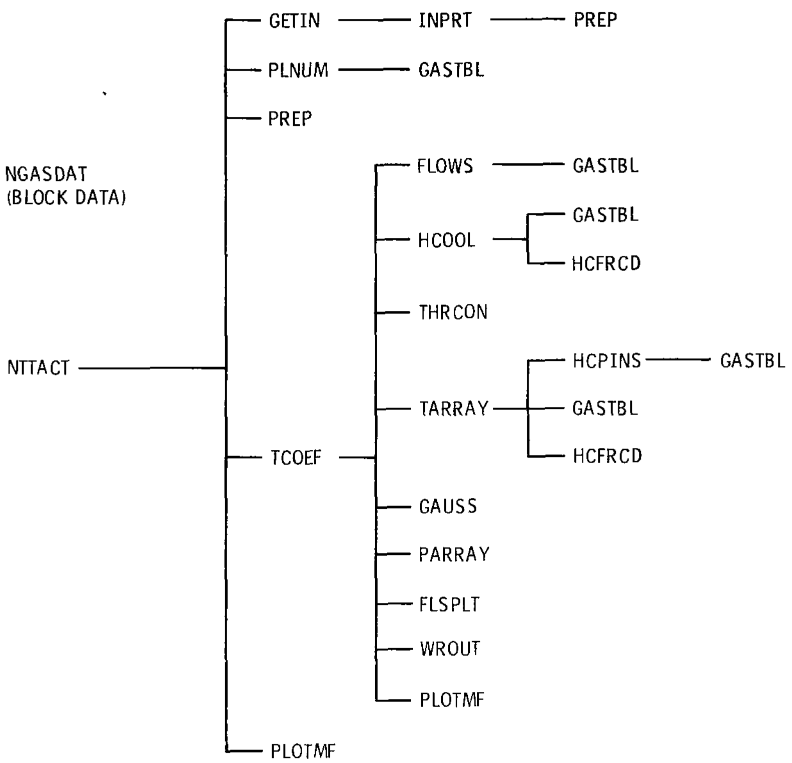


Figure 4. - Subroutine calling relations.

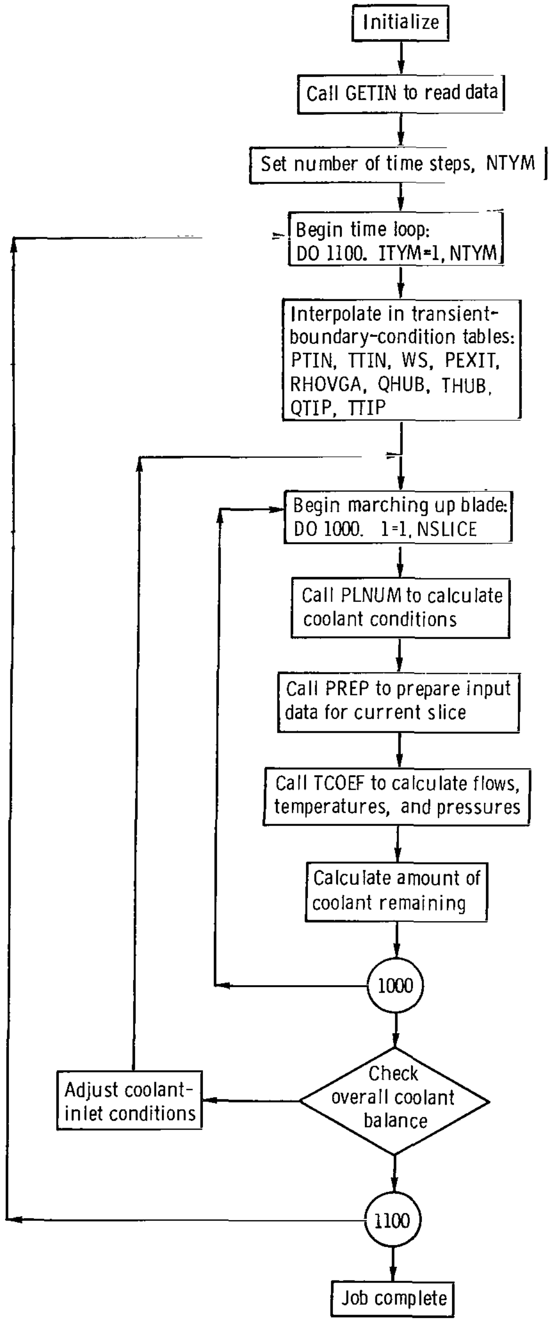


Figure 5. - Flow chart of main program.

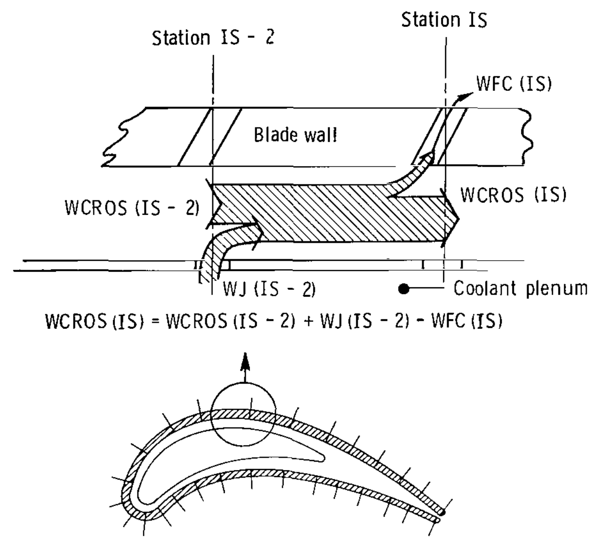


Figure 6. - Coolant-channel mass balance.

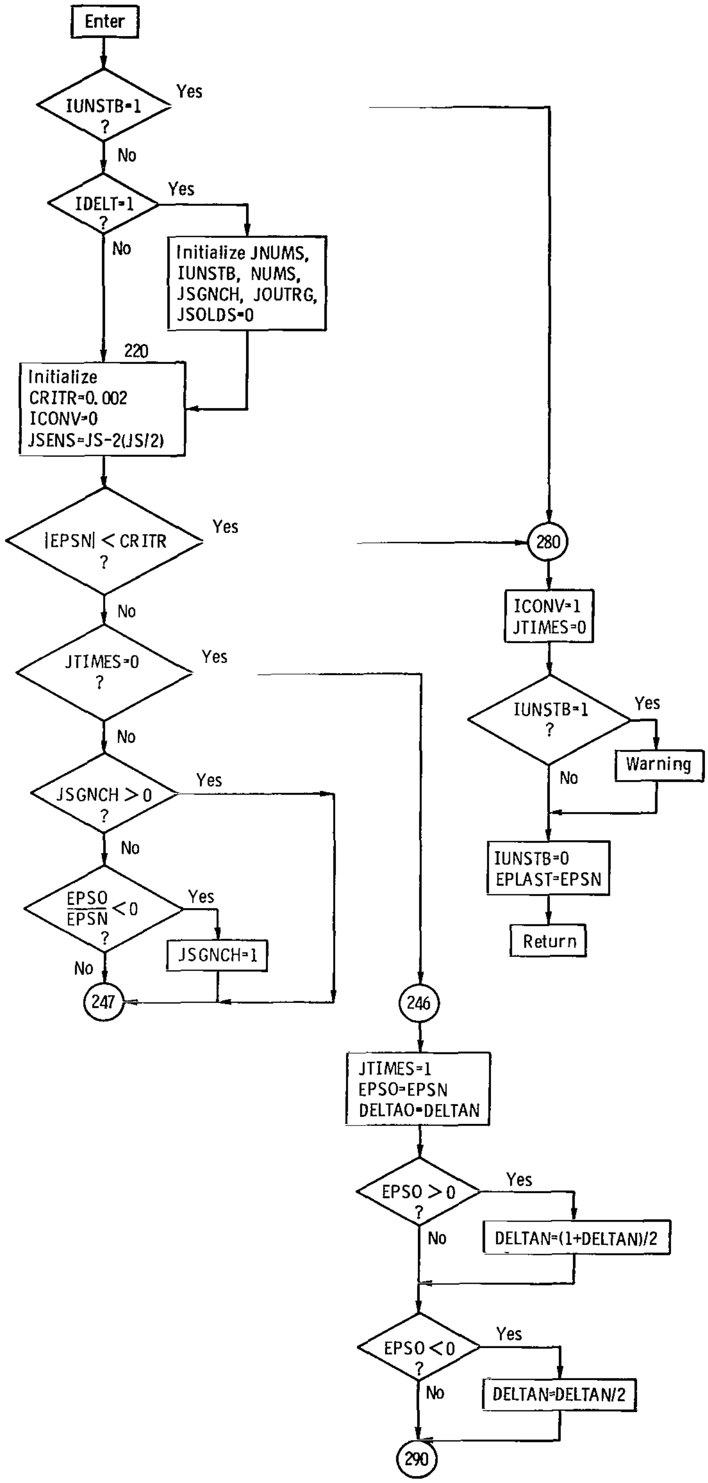


Figure 7. - Flow chart for subroutine FLSPLT.

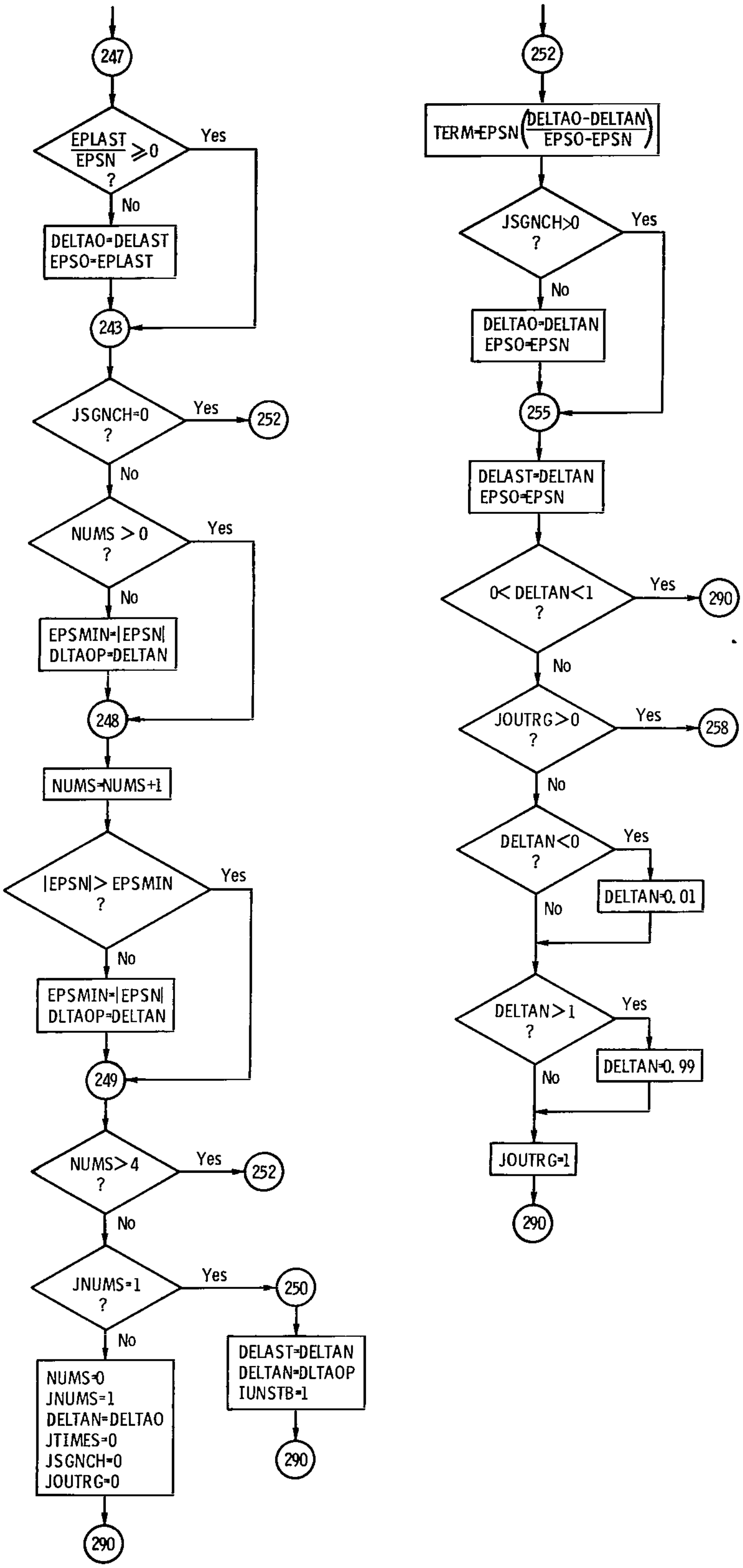


Figure 7. - Continued.

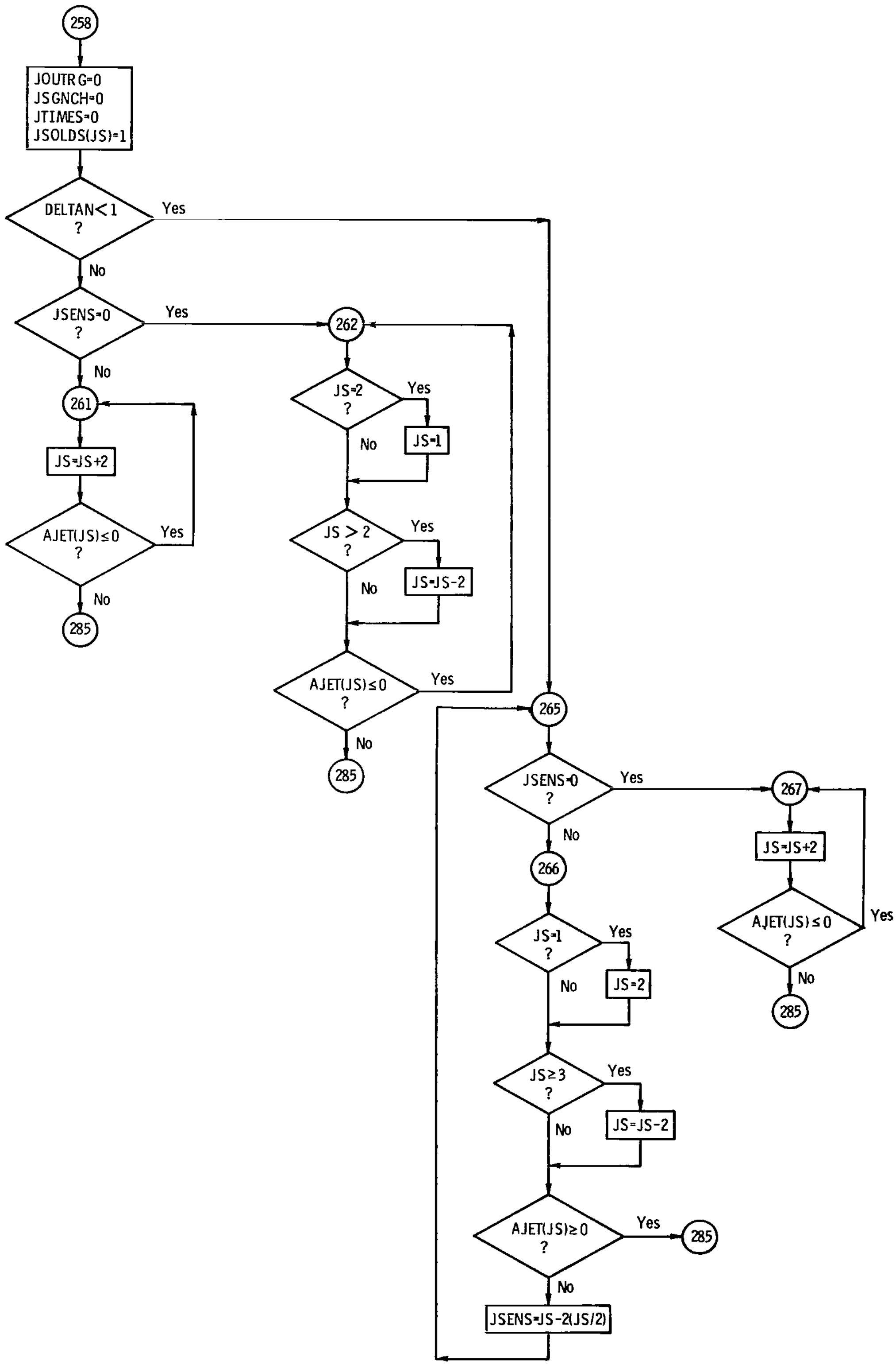


Figure 7. - Continued.

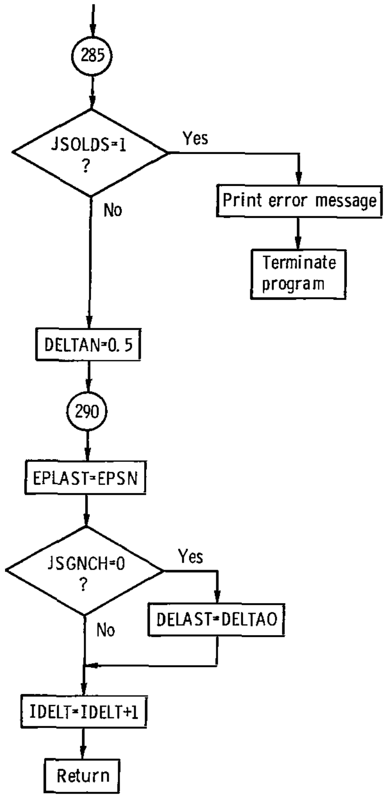
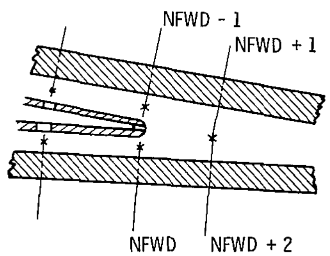
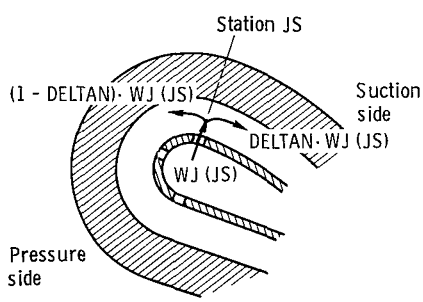


Figure 7. - Concluded.



(a) Detail of station numbering at end of impingement insert.



(b) Detail of flow-split point.

Figure 8. - Details of flow-split parameters.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-1391		

4. Title and Subtitle
TACT1, A COMPUTER PROGRAM FOR THE TRANSIENT THERMAL ANALYSIS OF A COOLED TURBINE BLADE OR VANE EQUIPPED WITH A COOLANT INSERT. II - PROGRAMMERS MANUAL

5. Report Date
January 1979

6. Performing Organization Code

7. Author(s)
Raymond E. Gaugler

8. Performing Organization Report No.
E-9767

9. Performing Organization Name and Address
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

10. Work Unit No.
505-04

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D. C. 20546

13. Type of Report and Period Covered
Technical Paper

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract
A computer program to calculate transient and steady-state temperatures, pressures, and coolant flows in a cooled axial-flow turbine blade or vane with an impingement insert is described. Coolant-side heat-transfer coefficients are calculated internally in the program, with the user specifying either impingement or convection heat transfer at each internal flow station. Spent impingement air flows in a chordwise direction and is discharged through the trailing edge and through film-cooling holes. The ability of the program to handle film cooling is limited by the internal flow model. Input to the program includes a description of the blade geometry, coolant-supply conditions, outside thermal boundary conditions, and wheel speed. The blade wall can have two layers of different materials, such as a ceramic thermal-barrier coating over a metallic substrate. Program output includes the temperature at each node, the coolant pressures and flow rates, and the coolant-side heat-transfer coefficients.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Heat transfer Turbine cooling Computer program Impingement cooling	Unclassified - unlimited STAR Category 34

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	163	A08

* For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1979

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

12 1 1U,D, 121878 S00903DS
DEPT OF THE AIR FORCE
AF WEAPONS LABORATORY
ATTN: TECHNICAL LIBRARY (SUL)
KIRTLAND AFB NM 87117

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return